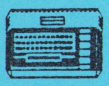
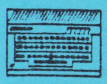
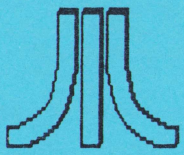
T — Y — N — E — & — W — E — A — R





TYNE & WEAR



ATARI 8-BIT USER GROUP

Newsletter of TWAUG

Software
Editorial
Buy & Sell
Hardware

TYNE&WEAR
ATARI
USER GROUP

Reviews
Help line
Section
Repair Info

Public Domain Library

ISSUE #14

MARCH/APRIL 1995

U — S — E — R — G — R — O — U — P

BRING YOUR EIGHT UP TO DATE with power products from COMPUTER SOFTWARE SERVICES

THE BLACK BOX

The BLACK BOX is an add-on board for the Atari 600XL,800XL and 130XE 8-bit computers. It is a T-shaped board that plugs into the PBI port of the XL computer, or the ECI and cartridge ports of the 130XE. Connectors for both types of computers are built into the BLACK BOX so no adapter boards are necessary. A cartridge port is available on the board itself for 130XE users.

The BLACK BOX provides many unique and useful functions. The four primary functions are:-

* RS-232 serial modem port
* Parallel printer port
* SASI/SCSI hard disk port
* Operating System enhancements

The BLACK BOX is $199.95 for the basic unit, and $249.95 with an onboard 64K printer buffer.
Shipping and Handling extra.

THE BLACK BOX ENHANCER

A must for all BLACK BOX owners. The BLACK BOX ENHANCER is a plug-in module for your BLACK BOX, enhancing the printer functions and adding an instantly available, full featured sector editor!

Installation of the BLACK BOX ENHANCER requires one simple solder connection. Only $49.95 plus shipping/handling.

THE FLOPPY BOARD

Our latest and greatest product. The FLOPPY BOARD is an add-on expansion board for the BLACK BOX interface. It allows the use of the same inexpensive floppy drive mechanisms used in IBM computers. The FLOPPY BOARD is the first floppy drive interface to support "high density" floppy drive mechanisms in either 5.25 inch or 3.5 inch. Built into the FLOPPY BOARD are our BLACK BOX ENHANCER and a version of our SUPER ARCHIVER to allow copying of protected disks for 3.5 inch format. Included with the FLOPPY BOARD is our program to read and write to IBM or ST formatted disks. This makes the FLOPPY BOARD the best way to transfer files to and from your 8-bit.

The FLOPPY BOARD is only $149.95 plus shipping & handling.

THE MULTIPLEXER

This device brings the power and flexibility of larger systems to your 8-bit. The Multiplexer is a collection of cartridge interface boards that allow up to 8 Ataris to read and write to the same drives (typically a hard disk), access the same printer(s), and talk to each other. It is the first practical networking system for the Atari 8-bit computer.
One "master" computer (any 8-bit) is equipped with the master Multiplexer interface.
Then up to 8 "slave" computers hook up to this master, each having their own slave interface. The "common" peripherals (things that are to be shared) are connected to the master. On each slave, all disk and printer I/O is routed through the master, so no extra disk drives are needed.

The Multiplexer sells for $199.95 for a master and two slave units with cable. Additional slave units are $89.95 each, plus shipping/handling.

THE SUPER ARCHIVER II

The SUPER ARCHIVER II edits and copies all enhanced density programs plus retains all the features of the SUPER ARCHIVER.

The SUPER ARCHIVER II is only $99.95 plus shipping & handling. NOTICE: if you already have THE SUPER ARCHIVER you may upgrade to S.A.II for only $29.95 plus shipping/handling. Software only.

THE BIT WRITER

The Super Archiver BIT WRITER is capable of duplicating even the "uncopyable" Electronic Arts and Synapse Syn-series, which employ 34 full sector tracks. The BIT WRITER must be used with the SUPER ARCHIVER

The BIT WRITER is only $79.95 plus shipping/handling.

THE ULTRA SPEED PLUS OS

The Operating System that should be in every XL/XE computer! The Ultra Speed Plus puts unbelievable speed and convenience at your fingertips.

Use any DOS to place Ultra Speed formats on your disks (with XF551 or modified 1050 drives), reading and writing at this speed with most programs. This high speed mode can be turned off for maximum compatibility.

Four simple solder connections are required for installation if your machine has a socketed OS ROM. The Ultra Speed OS is only $69.95 plus shipping/handling.

For more information on these and other 8-bit products:

CONTACT
COMPUTER SOFTWARE SERVICES
PO BOX 17660
ROCHESTER, NEW YORK 14617
USA

ORDERING LINE: (716) 429-5689
FAX: (716) 247-7158
BBS: (716) 247-7157

EDITORIAL

Who to blame!!!

John Matthewson
David Ewens
Max Gerum

As promised in the last issue I used a larger font, BUT, I had a miserable time with the layout of the pages with that font. I had to change all the configurations in Daisy-Dot and I couldn't get it right, so I reverted back to the font I had used in previous newsletters. But as you can see it is printed in standard print and not as before in high density, which was of course smaller.

The reason is simple, if I had carried on and persevered with the larger font the newsletter would probably not have been ready by mid-March and that would have been a disaster.

The contribution fees for home and abroad:

HOME	1 COPY	£2.00
-DO-	6 COPIES	£11.00
EUROPE	1 COPY	£2.20
--DO--	6 COPIES	£12.50
ELSEWHERE	1 COPY	£2.50
---DO----	6 COPIES	£14.00

REMINDER:

The book--The Complete and Essential Map--anyone unable to pay the full amount of the cash price can pay in instalments, whenever and whatever you can afford, at no extra charge.

The next issue will be ready by mid-May.

CONTENTS

EDITORIAL 3

LETTER SECTION 4

NOISY 1050
how to quieten down your drive
by Paul Hollins 8

GAME REVIEWS
by our new reviewer
Kevin Cooke 9

DISK CONTENT 11

ADVERT FOR the OL'HACKERS
USER GROUP NEW YORK 12

CODING CAPERS
educational programming 13

MARK'S GAME COLUMN
by Mark Stinson 15
MAPS for ZORK I 18
WORDSEARCH SOLUTION
has been inserted in Map
section due to lack of space 18

BASIC TUTORIAL-ANIMATION
by Ofer Saferman 22

POWER USERS' CORNER
by John Kasupski 26

WORD SEARCH PUZZLE 29

PEN REFILL FOR 1020 PLOTTER
by Bill Walraven 30

CRACKING THE CODE
by Keith Mayhew 31

MICRO DISCOUNT 39

ACPC 40

CURRENT NOTES & PHOENIX 40

LETTER SECTION

Dear TWAUG,

I read with dismay the article entitled Mega-review by Fred Meijer in issue 13 of your newsletter. To say the least it was not very flattering with regards to the software I either sell or have written. Now I can accept the fact that people have different opinions about software and that people have every right to say they don't like a piece of software, however, under these circumstances it is usual to give proper reasons for disliking a certain software title. However, it is not right for somebody to go around writing and printing lies about somebody else, in fact there are laws called 'Libel' that should protect against such events. I would like to take this time to go through the above mentioned article and point out some of the many inaccuracies and false statements as well as offering my points of view on some of the statements made.

The first comments are directed towards two old games Jawbreaker and Mousekattack, yes they are re-releases of old games, but how many times do you hear people say, "if only the copyright holders of the old games could arrange to have them sold again" or words to that effect. In short I feel that many newer users would love the chance to buy these older programs at todays prices. What Mr Meijer fails to mention is that you get two versions of Jawbreaker (on one disk) that was originally sold as separate games for nearly $30 each when they were first released, you can now buy them direct from me for only £4, also you can buy Jawbreaker (both versions) and Mousekattack together for only £6.95. The comment about Jawbreaker being an exact copy of Pacman is not true, sure Jawbreaker is based on a similar game style to Pacman but it is not an exact copy. When Jawbreaker was first released, Sierra On-Line (who first sold Jawbreaker) were taken to court by Atari for copyright infringement. Atari lost the case because their copyright only covered the graphic imagery. Does Pacman have a toothbrush? It's like saying Street Fighter II and Mortal Kombat 2 are exact copies of each other and because Street Fighter II came out first Mortal Kombat 2 is a load of cr*p! Which is complete nonsense. Just because a game is old it doesn't mean it is no good, just look at Star Raiders, many people still consider this the best game ever made and it is older than Jawbreaker. Even if a game is old, to somebody who has never seen it before it is new to them!

About the only similarity between Pacman and Mousekattack is that they are set in a maze! In Mousekattack you have to lay pipes throughout the maze whilst using the cats as blocks for the rats, note you cannot 'eat the cats'. Mousekattack is a thinking mans arcade game, the cat feature and simultaneous two player option lift it way above an ordinary maze game in terms of playability.

The statement, "I also believe that almost everyone has an illegal copy of these games in their collection," is in very poor taste. Surely these days people realise that piracy has seriously affected the life of the Atari 8-bit machines, and surely statements like these do not belong in print no matter how much truth is behind them. Just because a large number of people might have illegal copies does that mean that decent law abiding owners should be denied the chance to legally purchase such classic software?

Now onto Arena. There are a number of false statements in this section, firstly DGS do not sell Arena (or Bubble Zone), the game is published (and sold) by Visionaire Software and also distributed by Richard Gore and KE-Soft - the same is also true for Bubble Zone. Secondly I doubt

LETTER SECTION

very much the game was written in 'normal' Atari BASIC, I'll check with the author but I think it was written in Turbo BASIC and compiled for speed. Thirdly the original demo version published by Page 6 was also in compiled BASIC and it does run under 'normal' Atari BASIC (please note BASIC all uppercase as it is an acronym). If you disagree you are welcome to examine my copy of Page 6's issue 66 disk which I received with my regular subscription. Fourthly what makes Mr Meijer think the extra capabilities of Turbo BASIC have not been used and what gives him the right to pass comment on the knowledge of the programmer?

Now Bubble Zone, most of this section of the review just gives Mr Meijer's opinion, which he is entitled to do. However I object to the line 'This is not a game which can be sold commercially.', in fact this game has been sold commercially and will still be sold, in fact several people have bought it after seeing and playing it!

The final game in the review is Alien Blast, a game written by me exclusively for use with a light gun. Reading through this review made me very angry and quite upset. How would you feel if somebody wrote this about a game you wrote. "The game itself is obvious a type-in listing from some magazine. Even the aircraft which has to shoot the aliens and would normally be controlled with the joystick is still on the screen."?
Firstly the game is not a type-in listing, I wrote all the game code and designed the graphics myself. The only part I did not write was the music, and this is a PD listing which I used to add some music to the title screen. I did not hide this fact, it says so in the English manual and it also says so in the scroller at the start of the game. The first line of the above statement '..obvious type-in listing...' is simply not true, it is a lie and a personal insult to myself. Basically I am being accused of stealing other people's sofware - this is libel and against the law. Mr Meijer has no proof to back up this statement simply because none exists, it is a lie. Don't you people at TWAUG realise you are legally responsible for what you publish and by publishing lies like this I have a legal right to sue you and Mr Meijer for libel in a court of law?

The bit about the spaceship is nonsense, in the manual it states that you can use the Z and X keys on the keyboard to move the ship left or right to stop the downward travelling bombs from hitting it. This is a purposefully added feature of the game, simply to make the gameplay more challenging.

Mr Meijer goes on and says he has looked at my code, which I have been advised is technically infringing on copyright and is illegal.
Okay now I'll address the opinion parts of the review of Alien Blast. The comments about Operation Blood are a bit off target, sure Operation Blood is a good game, I bought the joystick version not long after it was released, what happened when I wanted the lightgun version? I was told I had to buy the new version for the full price even after I had bought the joystick version - hows that for service!!!! Needless to say I didn't bother. It was a decision I took not to put a cursor on screen, I thought it would make the game far too easy if you could see exactly where you were shooting and can see where your shot hit, a brief explosion appears on screen.

Actually I do have a manual for Turbo BASIC programming thank you very much and I am aware of the BGET command and indeed I used it to load the font files. I didn't use it to load the digitised speech because there is a simple USR call to the digitised speech routine to do the loading of the speech data. It's true I didn't take the time to compile the Turbo BASIC code simply because I am pretty sure that it wouldn't work. The speech is stored

LETTER SECTION

in the extra memory banks of a 130XE and I think when the bank switching occurs the majority of the game code is banked out of sight of the cpu and the game would crash. If anybody knows differently I would love to hear from them. Also the game worked fine to my satisfaction and I saw no real need to compile it. Mr Meijer I am not ashamed to sell this game, I am in fact quite proud of it. Several copies of this game have been sold at computer shows after people have had a chance of seeing and playing it.

On several occasions, after calling the software rubbish Mr Meijer suggest it should be made PD, does Mr Meijer think that all PD software is rubbish or that PD is another name for rubbish? If so I know a good few PD authors who will not be impressed.

Mr Meijer also has several remarks for Kemal of KE-Soft. I can't speak for him but I can say that it is certainly none of Mr Meijers business as to what sort of deal I have with KE-Soft as regards payment of royalties. While we're on the subject of money, I am not giving support to the Atari 8-bits just to make lots of money, far from it, I have a full time job which has nothing to do with computers and I make more money per week doing my job than I make in a year selling Atari 8-bit software. Programming and selling Atari games is my hobby and I enjoy doing it and I also know of several people who like my products and the support I give. I can accept that people like to express their opinions and have a right to do so but when people start writing and printing out and out lies about me, I really wonder is it all worthwhile?

Finally I would just like to say I am disgusted that TWAUG have allowed such trash and lies to be published. I thought they were giving support to the 8-bit but after this I don't think I want to be involved with them anymore. So thanks for running my adverts in the past but please don't run them anymore. I'm saddened to think in a time when we Atari 8-bit users are fighting for survival that letters like this are necessary, we should be pulling together and giving each other support not fighting like children but I am not about to stand by and let people trash my name.

Yours most disgusted, R.GORE

REPLY from David

Yes Richard, we do understand the upset the article from Fred Meijer has caused. I believe I myself would be angry if someone wrote a similar article about a program that I had produced, but I would have accepted that, as that person has a right to his personal oppinion.

You say that you are surprised that TWAUG should have published such lies, I do not think that it is our right to say that Fred was telling lies. You also say that even if it were true, we still shouldn't have published it. If we do not publish something because we think it is a lie, and if we do not publish something because we think it is true, then what do we have left to publish?

The subscriptions we receive from Fred Meijer and all our other subscribers goes to produce the newsletter and to help keep it going and we believe that any of our subscribers has the right to have his or her articles or letters published in the newsletter. If we were to start refusing items sent in by the subscribers, we would soon have no support at all.

You say that we can't really be supporting the 8-bit, I would be interested to hear from anyone else who thinks the same as you. I have been a dedicated 8-bit supporter now for 14 years, Max for 9 years and John 6 or 7 years. I receive phone calls and letters almost every day of the week from people asking for information and help, and I'm always glad to give what ever

LETTER SECTION

assistance I can. If I can't help, then I'll do my best to find someone who can.

We regret that you no longer want to have anything to do with TWAUG and that you want us to remove your add from the newsletter, we will respect your wishes.

We have asked Kevin Cooke (our new reviewer) to do a review of the same games as Fred Meijer and you will find his article elsewhere in this issue. As you will see, someone's dislikes can be another person's enjoyment.

Dear TWAUG

In issue 13, Andrew McIntosh raised several points and queries. He also made the disastrous mistake of invoking my name, usually enough to make me put down my beer and bang out a reply. So here goes with some answers and an opinion (I can never resist):

Dear Andrew,

First, DiskMaster will happily output disassembly to a disk file--just enter the filespec. I use the program frequently in that mode and find it very useful. The other programs I mentioned, Debug and Bbkmon, can be found on the ANALOG magazine disks. Debug was in issue #39 and Bbkmon somewhat later. Both can also disassemble to file, though with Debug, output is limited to D:filespec (you can't specify a different drive.

On cold start, the operating system most definitely does clear all RAM from $0008 to whatever it determines is the end of RAM (dependant on whether Basic or a cartridge is present). A run at $E477 should produce a blank screen until the disk boot is underway. If you're getting garbage on the screen, don't worry about it.

Accessing extended RAM on an XE compatible upgrade is not hard. As your upgrade allows access to the XE banks you should be able to get at the other ones without difficulty. Any problems you are having are undoubtedly attributable to software. I sent David an article on RAM bank control last year. Possibly your letter will nudge him to edit and publish it. In the interim, here's the most likely series of PORTB (54017) values you need for the twelve banks, starting with the 130xe banks:

$E3,$E7,$EB,$EF
$C3,$C7,$CB,$CF
$A3,$A7,$AV,$AF

In the event that your upgrade is patterned after the original Newell one, the "A"s in the last line above become "8"s. Note that the values shown are with the OS on (a necessity except under SpartaDOS) and BASIC off.

I can come up with a plausible explanation for your weird self-test results but it's far too long for a letter. You need to read the article I sent David first. In fact, I believe if you do so, your question will have been answered.

There was a version of Turbo Basic released for the Atari 400/800. Essentially, it incorporates DOS 2 as part of the boot file (i.e. DOS.SYS is over 100 sectors long). Because the DOS is included with the Basic it can't be used except as a boot file. However, Tom Hunt of Ohio, USA took the file and stripped the DOS from it to get it to work with SpartaDOS (TB32Q.COM) and his hack works quite well. Because the RAM under the OS ROM is not used, low memory is bumped quite a bit and a FRE(0) call will show about 20k as opposed to the 32k+ you normally get with Turbo. The 400/800 version was also released under the name FROST Basic (FRank OSTrowski, I assume).

I am aware of no release of the DOS 2.5 source code but the 2.0 source was printed by Compute! Books under the title "Inside Atari DOS" authored by Bill Wilkinson.

LETTER SECTION

OSS does not exist. All rights to their products were acquired by ICD several years ago and, last year, ICD sold them to Fine Tooned Engineering in California along with rights to all their other 8-bit stuff.

Now that I've answered as many of your questions as I can, I'll stick my neck out with an opinion: The fact that you asked for DOS 2 and 2.5 source code indicates to me that you are never going to get the best out of your 256xl. With any extended RAM Atari, you need a FMS which supports the extra memory to its fullest. In other words, switch to MYDOS or, if you want real power, to Sparta.

John Picken from Victoria, Canada

NOISY 1050

Written by Paul Hollins

The Atari 1050 is the most widely used disk drive on 8 bit systems. They are reasonably fast, store quite a lot of data, but they do suffer from one major drawback...they are incredibly noisy!

The time that this becomes the most apparent is when everyone is upstairs in bed, and there you are still hacking away at the keyboard on your latest masterpiece and it's time, once again, to do the dastardly deed and SAVE your work to disk. You carefully type SAVE "D:filename.ext" and gently press RETURN. CLUNK! WHIRR! BEEP! Congratulations you've just suceeded in waking the entire household.

So, what can you do about it? Well, you could buy an expensive air-tight, sound-proofed case to put your drive in. However, it's not exactly practical is it? Ok then, well you could try and get an internal modification kit to quieten it down, but they are expensive and very hard to get hold of. So what now?!

NOISY 1050

Fortunately there is an answer, and the good news is that you do not need any knowledge of electronics whatsoever, because no soldering is required and, here's the best bit, it's absolutely free!

All you need is a small phillips type screw-driver, a couple of cotton-wool buds (the type you use to clean your ears with), a jar of Vaseline and a relatively dust-free environment to work in.

GETTING STARTED

Remove any disks and unplug all the cables from the disk drive, including the I/O cable. Turn the drive upside down and remove the six small screws holding the upper case to the lower case. Without seperating the case turn the drive back up the right way. Remove the top case by lifting upwards from the rear of the disk drive. The black plastic front-panel can now be removed as well.

Without touching anything, study the inside of the disk drive and look for two small bars with a sliding 'thingy' in the middle of them. This 'thingy' is in fact the READ/WRITE head and now that you found what I'm talking about, I must point out that, obviously, extreme caution is needed.

Using a cotton-wool bud, apply a little Vaseline onto each of the two bars being very careful not to use too much or get any on the READ/WRITE head itself. Once you've done that gently move the READ/WRITE head and put some Vaseline underneath where this was first sitting, once again being very careful. By the way, you may think that the more Vaseline you use, the quieter the drive will be, and to a certain extent, you'd be right. However by doing that, you also run the risk of getting it onto the READ/WRITE head, and I always find that if you do use too much, you can't hear a reassuring purr from your drive, and then you start wondering whether it's functioning correctly or not.

NOISY 1050

That said, you now have a much quieter 1050. It's now time to re-assemble the drive. Refit the black plastic front plate and top cover, securing the six screws removed from the lower case. And finally, re-connect the drive to your computer and power supply.

Overall the process is quite simple and I can't see you having any problems with it whatsoever, but I must remind you that if you do attempt it you will invalidate any warranty you may have on your disk drive. Also I can not be held responsible for any damage caused whilst you are attempting to do, or resulting from, what I have suggested in this article. Also, you will find that you have to repeat the process, from time to time, when your drive starts to get noisy again. I usually find that I have to do this every 6 to 12 months, depending on useage.

Incidentally, some people prefer to use sewing-machine oil or 3in1 oil instead of Vaseline. It's just a matter of preference, as all of them have the same effect in the end.

GAME REVIEWS

By Kevin Cooke

Well, as you will probably know by now, I have accepted the task of being TWAUG's regular game reviewer - I hope you find my reviews up to the same high standard as Mark Fenwick achieved. If you have any queries, suggestions or requests for certain programs to be reviewed, let me know c/o TWAUG. However, for my first issue, the guys have asked me to give an English point-of-view on the games reviewed by Fred Meijer last issue, namely the software being sold by Richard Gore. I will go through each title and give my own point of view on them. Remember, my view isn't "more correct" than Fred's - it may be different but

GAME REVIEWS cont.

everyone has their own opinions and all are equally valid providing the information used to form them is correct.

Title: JAWBREAKER/MOUSEKATTACK
Sold at: DGS and Ke-Soft
Price: £6.95 or DM 24.00

Well, as Fred rightly pointed out last issue, both of these games are re-releases of two 1980's games. The two games (as part of a double pack) are packaged in a plastic sleeve and come with an A5 manual describing loading, gameplay, etc. These manuals are well produced - I particularly liked the story about how the Jawbreaker game originally had to be changed for release. However, the most important part of the package is the games, so what are they actually like? Well, both are variants on the Pacman theme. Jawbreaker is practically identical except the Pacman has been changed to a Jaw and the ghosts have also been changed. In Mousekattack, the object of the game is to travel over every part of the maze, laying pipe as you go.

Although the graphics are not as good as in Pacman (there is no real colour scheme giving the screen an "artifacted" look), the gameplay is thankfully still there and is better than any of the Pacman style PD games I have played. Come to that, it also fairs well against the commercial Pacman releases. As an added bonus, an extra maze was even added to Jawbreaker for it's re-release and, although it isn't really much different to the original, it was a nice thought.

Overall, I think Mousekattack and Jawbreaker are two good games - let down a little by the colour - scheme but good enough to keep you playing for a long while.

Title: ARENA
Sold at: DGS and Ke-Soft
Price: £5 or DM 19.80

I first saw Arena when it was given as a demo version bonus on the Page 6 issue disk. I must admit that

GAME REVIEWS continued

I liked the game even then and was very pleased when I saw Richard Gore selling the full version featuring many more levels.

The simple object of Arena is to guide a number of pods into the screen exit. However, the pods never stop moving so you will need to guide them directly into it by placing bits of wall in their way. If the pods touch certain parts of the scenery, they will die so it is important to keep them from doing this!

Generally, I think the graphics are quite well done for a Turbo BASIC game. Animation is used in the parts which can kill the pods and a good use has been made with colour. Sound is let down a little due to the lack of it! However, the best part of Arena has to be the gameplay as it can be really fun. The feeling of "just wanting to complete this level" is exactly the sort of feeling you would get while playing Lemmings. Overall, I really like this one. It can get very addictive.

Title: BUBBLE ZONE
Sold at: DGS and Ke-Soft
Price: £5 or DM 19.80

Bubble Zone is a shoot-em up style game in which the object is to guide your tank and shoot all of the bubbles. When the bubbles are shot, they sometimes leave behind bonuses, ie. extra energy, extra smart bombs, etc.

Graphically, Bubble Zone is quite good. Everything is large and fairly colourful and animation is used in the movement of bubbles, albeit a little jerky. There is also a nice title screen.

Unfortunately, although Bubble Zone starts off as fun and exciting, it does get a little repetitive quite soon and could have done with something in between levels to add some excitement. Some music playing in the background also wouldn't have gone amiss.

Overall, not bad but perhaps slightly over-priced.

Title: ALIEN BLAST
Sold at: DGS and Ke-Soft
Price: £4.95 or DM 19.80

So, finally we get onto Alien Blast, Richard Gore's own creation. But what is it? Well, alien blast is based on the classic space invaders game but, instead of moving a ship and shooting the invaders with a joystick, shooting is now done with a light gun.

In his review, Fred commented on the fact that the screen was obviously from a "type in listing from some magazine" because the aircraft "which has to shoot the aliens and would normally be controlled by the joystick is still on the screen". However, a quick read of the instructions would have told him that it can still, and must be, moved to avoid bombs which are dropped by the spaceship that occasionally flies overhead.

I would agree with Fred that a sight on the screen would have been nice and the ammo limit is too low but, as a plus point, it does make the game more difficult and therefore, a longer - lasting challenge.

One thing I must point out to Fred is that it doesn't make any difference whether or not Richard wrote a routine to do something that Turbo BASIC can already do. Also, the un-structured listing and the fact that it was not compiled also has no effect on the final game and so should not be included in a review of a piece of software.

Generally, I would say that Alien Blast is a reasonable game (although better on the 130XE when you also get digitized sound). Yes, I would have liked a greater ammo supply and possibly sights on the screen but, as it stands, it's still not bad. For light-gun users looking for something new I would certainly advise taking a look at it - the price won't break the bank after all.

GAME REVIEWS continued

Well, that's my overview of last issue's programs out of the way. To finish off, I will take a quick look at one other piece of software:

VIDEO CLASSICS

First of all, the name "Video Classics" is totally representative of what this software is - a revamped collection of 6 classic games. But don't let that put you off, these are really quite good.

The five games featured are Tennis, Football, Squash, Solo Squash, 4 Bat Flip and Asterbliperoids. In case you still haven't guessed what these games are, they are basically variations on the classic bat and ball games which you may have had on one of those old games machines which used to be popular. However, these aren't simply straight conversions with the plain graphics of the old versions. No, these have been vastly souped-up on the audio-visual front to produce some very good effects. For a start, the bats have been slightly changed to make them look a little more interesting. Also, instead of plain coloured backgrounds there are now moving patterns in three or four colours! Very funny on the eyes! Sound is also improved with an interesting tune playing in the background. So what are each of the games like?

Well, Tennis is your basic bat and ball game with a bat at each side of the screen and a ball whizzing between them. Football is similar except two bats are controlled to defend your own goal and score in your opponent's. Squash is played just like the real-life game with both bats on the same side of the screen in an attempt to hit the ball against a wall opposite them. 4-Bat Flip again puts you in control of two bats but this time one is at the side of the screen and one at the top - tricky! Finally is Asterbliperoids (bless me!). This is the same as the tennis game except for a few exciting differences - a space scene in the background and various moving planets and asteroids to reflect the ball back to you at a moments notice! This game almost looks like a completely different game from the others!

Overall, this is a winner. Classic games coupled with good graphics and sound, plus the option of playing against the computer or a human opponent. However, you may have noticed that I haven't yet mentioned where this can be bought from - that is because I don't know! You could try contacting Dean Garraghty as he may have it included in the stock of his used software but, as far as I am aware, the best way to get this software is to look out for it at a computer show (such as the All Micro Show) or to try and find it being sold second-hand. However, don't let that put you off as it's a great piece of software when bought at a budget price and worth the effort of finding.

Well, I hope you found this column interesting. Next issue I will go back to the normal style of reviewing and take some more in-depth looks at a smaller amount of games past and present. See you soon!

DISK CONTENT

Side A of this issue 14 disk has another good selection of programmes for you to enjoy.

AMISAMP.COM - this is a demo with a scrolling line and two photos on the screen. The only trouble, I find with the scrolling line is, if you don't know the German language you can't read the scrolling line, but you can still listen to the music.

LOTTERY - now there's a nice little program for you, that is if you enter your pounds worth of Lottery ticket every week. This program selects six numbers randomly for you for the lottery ticket. Norman Williamson is warning you not to take it seriously, he only wrote this program for a bit of fun.

DISK CONTENT continued

Look out for more goodies in future TWAUG issues from Norman Williamson.

AUTOTAP1.BAS - AUTOTAP2.BAS - these are two programs for the tape users, the documentation file is on the disk for you to read, under the filename of AUTOTAP.DOC.

SCRMMAT.BAS - Scramble Mat is a board type of game, you are presented with a 16 square full of characters which have been scrambled by the computer and you aim is to arrange them in alphabetical order. I found it quite difficult to do, the computer is timing you at the same time.

DOMINO.BAS - this is an educational maths program. It gives you the options to choose addition, multiplication, or subtraction, but instead of seeing numbers on the screen it displays dominoes and that may appeal to the children better.

DIRPRINT.BAS - this will printout a directory from a basic disk, instead of searching through all your disks for a certain program why not make a hard copy and file them.

ACCOUNTS.BAS - I am afraid I am unable to tell you anything about this program, to find out more about it load it and GOTO 30020 and be ready with the CTRL+1 to stop it from scrolling.

On this side of the disk is also a listing to go with Andrew Thompson's article.

Now we come to side B of the disk.

Here we have a game called Note Invaders, the aim of the game is to shoot the musical notes, instead of the usual aliens. You use the keyboard instead the joystick, but you must press the key that represents the musical scale. For instance if the note is an A you must press the A key, got it??? If you don't know the music scale don't despair, you have the option to see the music scale on screen by selecting that option in the menu before starting the game and the instructions can also be read by displaying them on screen. This program first appeared on a Futura newsletter disk and it is by Kevin Cooke.

FILINF12.COM - no don't say it "no not another database program", I am afraid yes it is but with a difference. This database gives you all the command option on screen and it also gives you the option to format a disk without leaving the menu. Now the difference of this program is it gives you plenty of space to enter your comments. You also got a very good search function.

There are five demo programs on this side side of the disk to go with the Animation program in this issue.

We have included the DISKCOM32 utility to be able to Un-Diskcom the ICONS.DCM file. These are a few icons by Ralph Bradley, to add to your PrintShop Icon collection.

THE OL'HACKERS ATARI USER GROUP INC.

O.H.A.U.G. is an all 8-bit user group in the State of New York, they are producing a bi-monthly first class informative newsletter on disk.

The disk is double sided full of news, views articles and bonus games and/or utilities. The disk has its own printing utility which you can use to read the content of the disk on screen or make hard copies. A large PD Library is also available.

For more information on how to join why not contact the President of this first class Atari User Group by writing to:

A. Pignato
O.H.A.U.G.,3376 Ocean Harbor Drive, Oceanside, N.Y.11572, U.S.A.

CODING CAPERS

by Andrew C. Thompson.

EDUCATIONAL SPACE-INVADERS!?

Welcome back to another nail-biting episode of your favourite caper, this time we're going to learn how to defend the world from a hoard of mean aliens with just a player and a missile. You got it! We're going to be blasting aliens by the numbers, but not only that! Whilst you can learn a few techniques as to alien blasting, your young pro-creations can get educated in maths.

Right then, the listing for this issue's column has been put on the disk. As you will see if you've loaded it into Turbo Basic, the listing has a number of steps:

1. INITIAL SETUP,
2. SHAPING OUR PLAYER,
3. BRINGING ON THE ALIENS AND
4. GAINING CONTROL OF EVERYTHING VIA THE MAIN LOOP.

So, without further ado let's describe the listing by the steps:

1. INITIAL SETUP.

Line-130 protects 16 pages (16*256=4096bytes) for our PMG's, line-140 selects our playfield, lines 150-160 calculates our PMG map (MEM) and the top of our playfield memory (DM). Lines 180-230 setup our text window colour, player/missile memory requirements and points to it, turns the PMG's on, prioritizes the players over playfields and makes the missile that we're going to use take it's colour from location 711, sets the PM sizes and selects PM colours.

The variables at lines 250-270 are:

XPG= Xco of player graphic
YPG= Yco of player graphic
AY= Aliens Yco decrement delay
AQ= Alien quantity
XPM= Xco of players missile
YPM= Yco of players missile
YMF= Yco missile in motion flag
PG= Start of player graphic memory
PM= Start of players missile memory

2. SHAPING OUR PLAYER and
3. BRINGING ON THE ALIENS.

Firstly we clear any unwanted garbage in the player and missile memory at lines 310-330, lines 380-410 POKE in the shape of our player, whilst line 340 and lines 450-510 POKE in the aliens, the target alien and prints up the maths question to which alien should be blasted. Lines 480 and 510 ensure that, although the aliens are random, the correct answer to the question 3+5 is placed on screen once only.

4. GAINING CONTROL OF EVERYTHING.

OK then, since we've setup everything, we now need to pass control of the player to the joystick, bring the aliens down at a governed rate and engage the use of our missile. This is all done using boolean style programming in the main loop. There are 2 reasons why I've used this style as opposed to multiple IF/THEN conditions: the first reason is because it is usually faster to process, but the more important reason is because it makes gameplay evenly timed.

For example, if I changed line 550 to it's IF/THEN equivalent:

```
550 S=STICK(0)
551 IF S=07 THEN XPG=1
552 IF S=11 THEN XPG=-1
```

You should realize that pushing the joystick to the right the IF/THEN condition on line 551 will fully execute, and the same applies to line 552 if I pushed the joystick to the left. But, what if I didn't push the joystick either way? Then neither of the IF/THEN conditions will execute! But what's the difference I hear you say? The difference is that the BASIC language can execute these 3 lines at different speeds, and if we have several IF/THEN conditions then the motion of your space-ship along with the interraction of the movement of the aliens and firing of your missile will all result in things moving at different speeds. Sometimes your spaceship will move fast, sometimes it will move slowly!
With this boolean method, careless if you move the joystick or not, careless if a missile is being moved

CODING CAPERS

up screen the BASIC language will process lines 550-730 at a constantly equal time because all the formulas (Boolean IF/THEN structures) are processed all of the time!

OK then, you might like me to describe a couple of the Boolean calculations for you, as upon immediate notice they do look a little mean don't they!

The JOYSTICK MOVEMENT is controlled by XPG=XPG+(S=7)-(S=11). Basically S can either be not equal to 7 or 11, or equal to just one of them, so IF S=7 the formula would return: XPG=XPG+1-0 etc..

The MISSILE VERTICAL MOVEMENT is controlled by the variables: F, YMF and YPM in lines 570, 580, 600, 650 AND 660. Thus:

```
570 F=STRIG(0)+(YPM<90)
580 YMF=(NOT F)+(YPM<90)
600 YPM=YPM-YMF+92*(YPM<0)
650 POKE PM+YPM,YPM<90
660 POKE PM+YPM+5,0
```

Basically, the starting position of the missile was set in YPM on line-260 to 90. Now, line-570 translates: F=STRIG(0)+(1 IF YPM<90, ELSE 0). Since, YPM=90 and is not less than 90 the formula of F=STRIG(0)+0. So, this means F=1 if you haven't pressed your button, but F=0 if you have! The reason the (YPM<90) is appended to this F formula is to stop the button being detected (F=0) once the missile has been launched, so that you have to wait for the original missile to have cleared the screen before you can launch another. YMF is the speed of the missile travelling up the screen. If the firebutton has not been pressed then YMF=0, but if it has YMF initially becomes YMF=1+0, the 2nd time through the loop, F is prevented from being 0 until the missile is done with, so YMF=0+(YPM<90). Now, since YPM is decremented in line-600, YPM is less than 90 and YMF hence becomes YMF=0+1, a value of 1 again! Thus, so far we can gather that YMF always equals 1 when the missile is in vertical motion. So, YPM=YPM-YMF on line-600 is always decrementing so long as our missile has been fired. But, what about stopping the missile when it has gone too high! Well, that's where the 2nd half of the YPM formula comes in. The 92*(YPM<0) segment of the YPM formula resets YPM to equal 90 when the last value of YPM is less than 0, and lastly to stop the missile from being rePOKEd at the starting position making things untidy, we only POKE a value of 1 in on line-650 if YPM<90!
Lastly, since the method of missile movement is to set bits in the missiles vertical co-ordinates, we also zero these bits 5 scan-lines lower on line-660, thus making the size of our missile.
It can be tricky understanding these formulas or making your own, but a little bit of determination and you'll get there.

The MISSILE HORIZONTAL MOVEMENT is mainly controlled with variables: CZ, XPM and XPG on lines 590, 610 and 640. Basically, XPM remains at the last position the missile was fired while the missile is in motion, but is changed to equal XPG+2 when the firebutton is next legally detected.

The last of the formulas are those for the aliens coming down and the missile to playfield collision detection between lines 670-730. The way in which the horizontal position of the fired missile is converted to a screen column is with the formula on line-700. Just take 47 off XPM and then divide by 8.

Anyway, that about concludes the topic of this issues capers, if you'd like more information on this Boolean style of programming or PMG's then you can hunt down an article I wrote for a back issue of TWAUG or get the Complete & Essential MAP. Anyhow, don't forget now... If your having a few problems with any type of programming, a little time and patience works wonders, the lending of another persons mind on the subject is very good too! Chow for now and good programming.

MARK'S GAME COLUMN

ZORK I --- Review by Mark Stinson

This issue begins a series on the Zork adventures, by Infocom, the masters of adventure. The series may not follow in each issue, as I have not yet completed Zork II, having only recently obtained the three Zorks. I say the 'three' Zorks, although there were actually eight, though not entitled Zork. The series went Zorks I-III, Enchanter, Sorcerer, Spellbreaker, Zork 0, and Return to Zork. The latter two were not released on the 8-bit.

Much has been written on Zork, indeed, it even spawned a book of the game! However, there will be many Atari users who have bought their machines relatively recently who may not have come across it. Well, it all began with a huge mainframe game called Adventure, by Crowther and Woods. Marc Blank and Dave Lebling co-authored Zork on a mainframe in 1977, and later worked together at Infocom to code the game in three sections, Zorks I to III for various home computers. The game was written in ZIL (Zork Implemetation Language) and set the standard for all future computer games. The series is still recognised by many as the ultimate in adventure with a good vocabulary and extensive prose.

Many of you will never have tried an adventure, or will have had a go at some poor game and given up in disgust. I can only say give Zork a try, it was for many the sole reason for buying a home computer. At the moment Page 6 have a supply of Zork I at under three pounds - that has to be worth a go.

Zork is set in a huge underground empire which has long been abandoned by all but a few colourful characters. Your aim in Zork I is to collect a number of fabulous treasures dotted about the empire. In order to do this you must first solve some truly diabolical puzzles, and learn to apply logical thought to difficult situations. I do hope that you will give this classic game a go, and for those who need help, the rest of this article is devoted to some of the trickier parts of the game.

HOUSE and OUTSIDE

<u>How do I open the egg?</u>

Give it to someone more skillful than yourself (later in the game).

<u>How do I open the grating?</u>

From the inside!

<u>How do I get inside the house?</u>

Open the window on the east of the house.

<u>How do I get the bauble from the songbird?</u>

Take the contents of the egg and activate it.

<u>How do I get on the rainbow?</u>

Look for something colourful later in the game, then wave it at the rainbow.

<u>How do open the door in the kitchen?</u>

Solve the puzzle of the Cyclops later on, and you will see.

<u>How do I get underground?</u>

Pull back the rug to reveal the trap door.

CELLAR

<u>How do I get past the troll?</u>

Use the sword, and hope for the best.

<u>How do I beat the thief (if he turns up here or elsewhere)?</u>

When you have enough points he is not too difficult.

MAZE

<u>How do I get through the maze?</u>

MARK'S GAME COLUMN continued

Look at the map in this article.

What do I do with the rusty knife?

Leave well alone.

How do I beat the Cyclops?

He is not proud of his father, and hates to hear the name of the hero that beat him (ODYSSEUS).

OTHER AREAS

How do I get the platinum bar?

Type ECHO.

How do I cross the lake?

Touch the mirror.

How do I cross the river?

You need a dingy and something to inflate it.

How do I enter the Dome room?

Use the rope.

How do I enter Hades?

By lighting some candles, ringing a bell, and reading a book!

How do I open the dam?

Press the yellow button in the control room then undo the bolt with the wrench.

How do I get past the bat?

Carry something smelly! Garlic?

How do I get light into the machine room?

Use the basket.

What do I do with the machine?

Find some fuel.

MARK'S GAME COLUMN continued

Is the shovel useful?

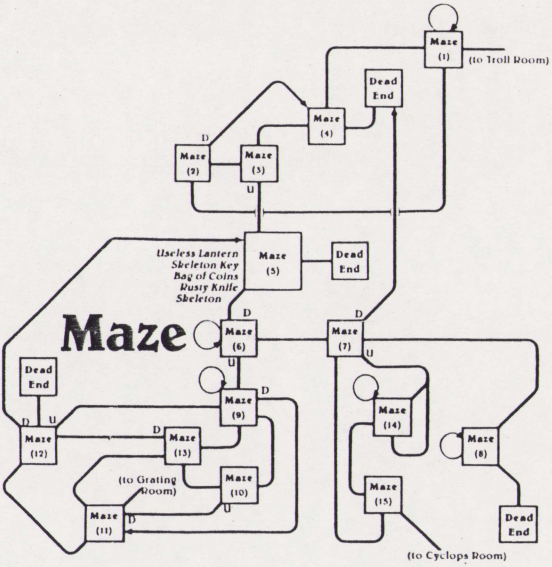
Dig in the cave.

TREASURE AND POINTS LIST

TREASURE	POINTS	POINTS FOR PUTTING IN CASE
Egg	5	5
Canary	6	4
Painting	4	6
Bar	10	5
Torch	14	6
Coffin	10	15
Sceptre	4	6
Trunk of Jewels	15	5
Trident	4	11
Jade	5	5
Bracelet	5	5
Diamond	10	10
Coins	10	5
Skull	10	10
Scarab	5	5
Emerald	5	10
Chalice	10	5
Pot of Gold	10	10
Brass Bauble	1	1

P.T.O. for THE MAPS

MARK'S GAME COLUMN continued



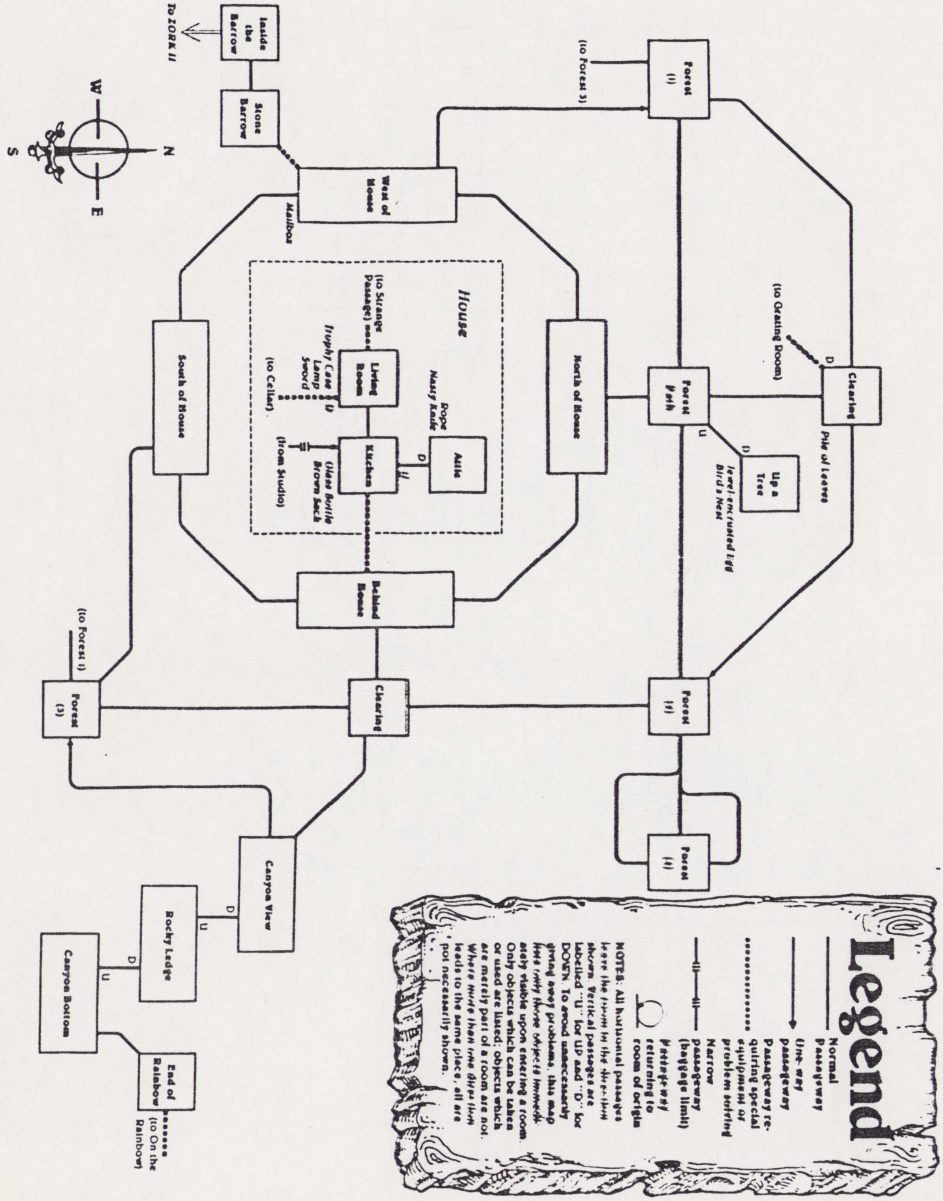
WORD SEARCH SOLUTION

·	·	N	A	M	P	M	U	J	·	·	·	·	S	A
·	·	C	E	T	Z	A	·	·	·	·	·	C	I	·
·	·	·	·	·	D	·	·	·	·	·	R	N	·	·
·	·	·	·	·	E	·	·	·	·	A	R	·	·	·
·	·	·	B	·	A	·	·	·	B	E	·	·	·	·
·	·	·	A	M	D	·	·	B	V	·	·	H	·	·
·	·	·	L	I	L	·	L	A	·	·	·	A	E	·
·	S	·	L	D	I	E	C	·	·	·	·	R	G	·
·	E	·	B	N	N	S	I	R	T	E	T	D	A	·
·	I	·	L	I	E	T	·	·	·	·	·	B	P	A
·	N	·	A	G	·	·	L	·	·	·	·	A	M	J
·	O	·	Z	H	·	·	·	U	·	·	·	L	A	N
·	O	·	E	T	·	·	·	·	K	·	·	L	R	I
·	G	·	R	·	S	P	I	D	E	R	M	A	N	N
·	·	·	·	M	O	O	N	M	I	S	T	·	·	·

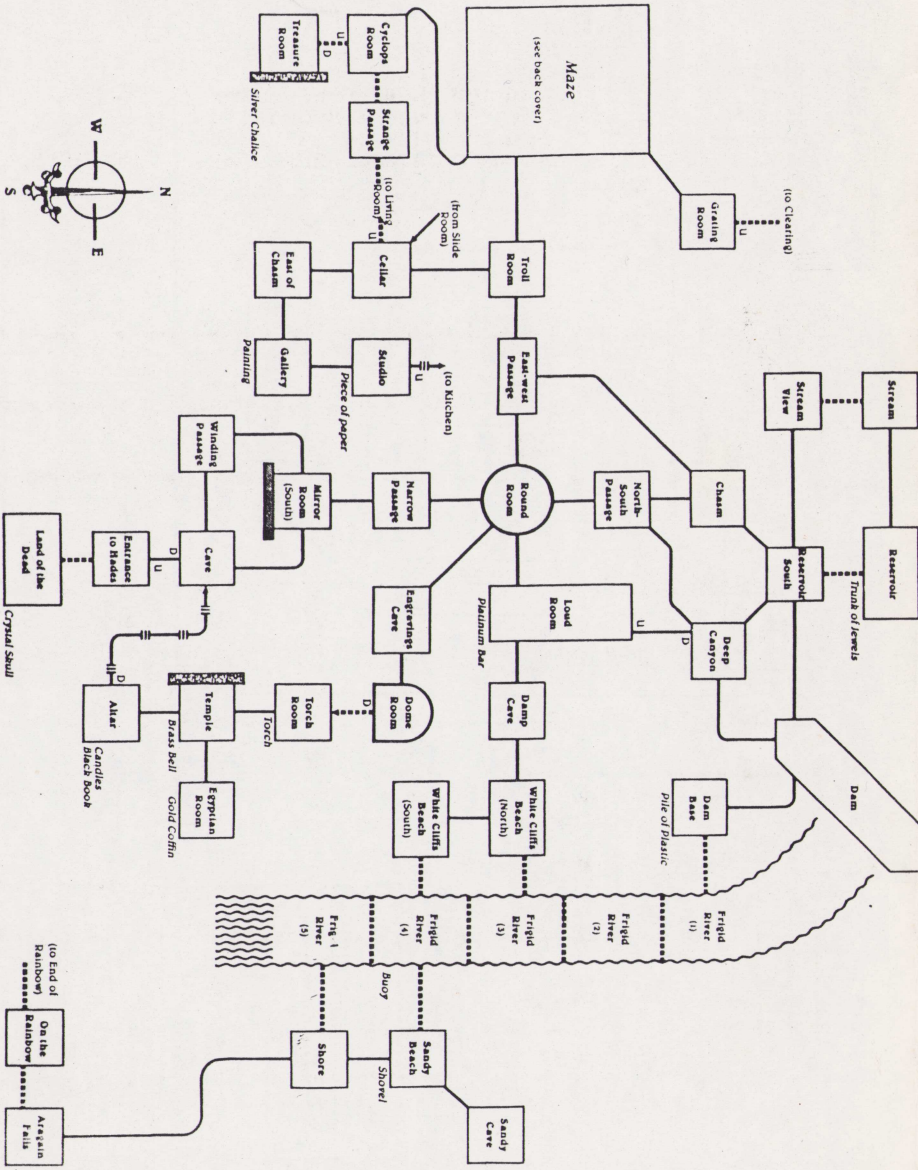
BALLBLAZER SPIDERMAN CAVERNIA DEADLINE HARDBALL
MIDNIGHT MOONMIST SCRABBLE JUMPMAN RAMPAGE
GOONIES TETRIS AZTEC NINJA KULT

TWAUG NEWSLETTER

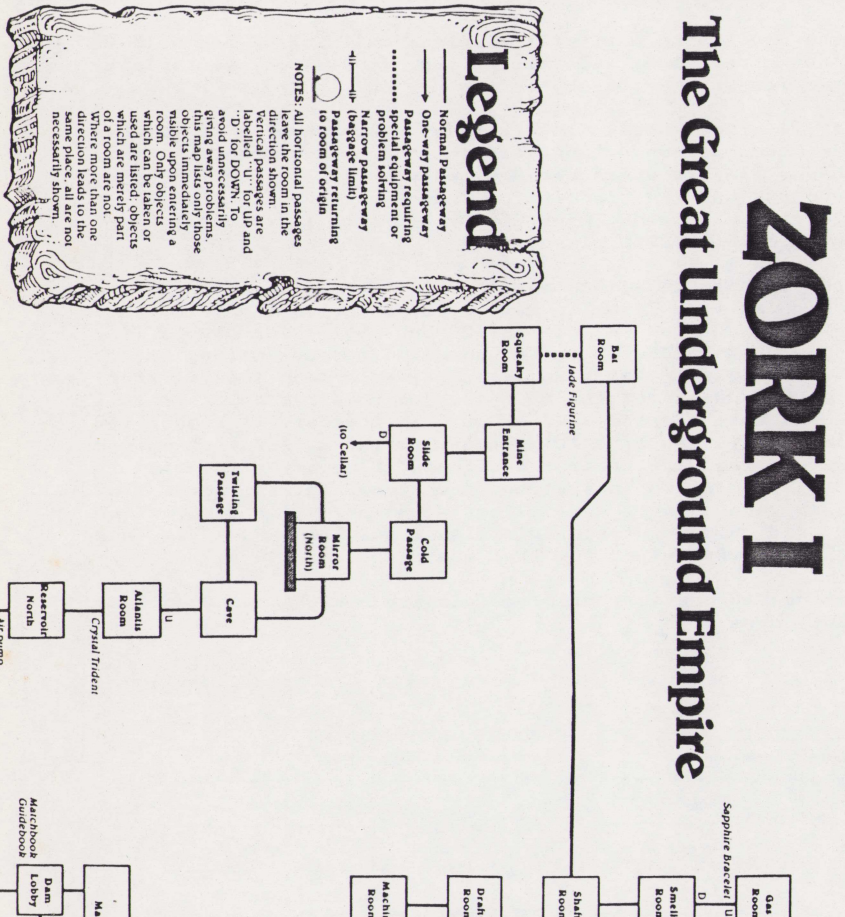
MARK'S GAME COLUMN continued



MARK'S GAME COLUMN continued



MARK'S GAME COLUMN continued



BASIC TUTORIAL - ANIMATION 4

By Ofer Saferman.

This is the concluding part of the animation tutorial and will be the most comprehensive and interesting. It will be about Player Missile Graphics or PMG for short from now on.

Atari has a great built in feature which enables it to move objects on the screen without erasing the background. The Atari computer has 4 players which are 8 bytes wide and 128 or 256 bytes high depending on the resolution chosen, and it also has 4 missiles 2 bytes wide and 128 or 256 bytes high. The four missiles can be combined to a fifth player.

Remember the problem to allocate memory for character sets that we discussed in article 2, well, here we have the same problem only bigger. First we need either 5*128=640 bytes for double line resolution or 5*256=1280 bytes for single line resolution, right? wrong. The Antic chip has a register called PMBASE (54279) where it holds the address of the PM base. For some reason it insists that the PM base address will be on a 1K boundary (4 pages) for double line resolution or a 2K boundary (8 pages) for single line resolution. Re-read the segment dealing with memory allocation in the second article on animation and then return here. The principle is the same, you allocate 4 pages for double line resolution and 8 pages for single line resolution and still take into account the unsafe portion that was discused.

After sorting the memory problem lets see how the computer sees the PM memory area:

```
SINGLE LINE RESOLUTION                  ¦ DOUBLE LINE RESOLUTION
======================                  ¦ ======================
PMBASE+2048--->¦-------------------¦ ¦ PMBASE+1024--->¦-------------------¦
               ¦      Player 3     ¦ ¦                ¦      Player 3     ¦
PMBASE+1792--->¦-------------------¦ ¦ PMBASE+896---->¦-------------------¦
               ¦      Player 2     ¦ ¦                ¦      Player 2     ¦
PMBASE+1536--->¦-------------------¦ ¦ PMBASE+768---->¦-------------------¦
               ¦      Player 1     ¦ ¦                ¦      Player 1     ¦
PMBASE+1280--->¦-------------------¦ ¦ PMBASE+640---->¦-------------------¦
               ¦ Player 0          ¦ ¦                ¦      Player 0     ¦
PMABSE+1024--->¦-------------------¦ ¦ PMABSE+512---->¦-------------------¦
               ¦ M3 ¦ M2 ¦ M1 ¦ M0 ¦ ¦                ¦ M3 ¦ M2 ¦ M1 ¦ M0 ¦
PMABSE+768---->¦-------------------¦ ¦ PMABSE+384---->¦-------------------¦
               ¦     768 Unused    ¦ ¦                ¦     384 Unused    ¦
               ¦  Memory locations ¦ ¦                ¦  Memory locations ¦
PMBASE-------->¦-------------------¦ ¦ PMBASE-------->¦-------------------¦
    (VALUE AT 54279 * 256)          ¦      (VALUE AT 54279 * 256)
-------------------------------------------------------------------------
```

BASIC TUTORIAL - ANIMATION 4 continued

Not as with character sets, with PM we have to do some heavier setup. But do not worry we will know it all by the time we finish. To generate PM graphics two chips cooperate: the ANTIC chip and the GTIA chip. We have to let them know that we want PM so there are 2 registers, one for each. The first is GRACTL (53277). If it is 1 then you will have missiles, if it is 2 then you will have players. If it is 3 then you will have both (logical isn't it ?). The second is DMACTL (559). If it is 46 you get players in double height resolution. If it is 62 you get single line resolution.

Of course we must not forget to store the value of the allocated memory in PMBASE (54279). With PM there is one more important thing, we have to clear the whole PM area, so when we display them on the screen we won't get any garbage. It is possible to do all that in basic but it is very slow and tedious, so a small machine language routine has been provided to do the task. Now that we did all that we are ready to design our players. They are designed in the same way as a character set but instead of a grid 8*8, we use a grid 8*128 or 8*256. We also must decide which player we want to design and thats the purpose of the table above. When you design a player you don't have to use all of it. You need to use only an 8*10 grid and position the player where you want.

So far so good, but we said that PM move so lets discuss movement. To move horizontally there are 8 horizontal position registers (HPOSP0-HPOSP3, HPOSM0-HPOSM3). These registers start at 53248 and go onward till 53256. Vertical movement is more complicated since a player by definition takes up the whole screen, so if you have designed a small shape and want to move it up or down, you actualy have to copy the data upwards or downwards in the player. Again this is easily done in basic but it is extremely slow, so 2 machine language routines have been provided, one for each direction. Now we can set them up, we can move them in any direction, but what about some colours ? Well, thats easy. There are 4 colour registers, one for each player.

The colour of the missiles is the colour of their respective player. There is a posibility to give all the missile a fifth colour, about that later. Another important thing about PM is that they have 3 sizes: normal, double, or quadruple width, and guess what, there are 5 registers, 4 for the players and 1 for all missiles. There are also some weird beasties called priorities, which define what will be seen in front of what you see, so for example if you don't like you PM's on the screen, you can have them behind the screen. The priorities are controlled using PRIOR (623) and are defined by telling the computer which colour comes first, (A table will follow), thus we can make PM appear in front or behind objects, almost as we please. The most important feature of our PM's is the computer's ability to detect collisions between them and other PM's and the playfield. This is very useful for arcade games to detect when someone got hit, or combined with priorities it can create nice effects, like a man circling a house, etc.

BASIC TUTORIAL - ANIMATION 4 continued

Am I bombarding you with a lot of dry boring information which doesn't necessarily explain who that horrible man is that made you read this article! But as I always say, the most important thing is to experiment with the demonstration programs, there are four of them on the issue disk. Just play with them, try to change all kinds of parameters and values. No damage can be done and should the computer lock up, reboot the disk.

Now for the bonus prize, because you have been so patient with me. Using PM's can achieve very versatile animation, in addition to the movement they can also change shape by page flipping inside them using a simple copy routine, so the PM can animate while standing still, or while on the move. Why do I say it's the bonus, because I am going to add a long machine language routine special written to be used in basic programs, which can do anything with player missile, and I mean anything, including the animation we just discussed. There will be a separate documentation for this utility with the programs.

Now there will follow a list of tables summarizing most of the information given in this article:

MISCELLANEOUS PM REGISTERS
==========

REGISTER	EFFECT	LOCATION
COLPM0	PM 0 Color	704
COLPM1	PM 1 Color	705
COLPM2	PM 2 Color	706
COLPM3	PM 3 Color	707
HPOSP0	Player 0 position	53248
HPOSP1	Player 1 position	53249
HPOSP2	Player 2 position	53250
HPOSP3	Player 3 position	53251
HPOSM0	Missile 0 position	53252
HPOSM1	Missile 1 position	53253
HPOSM2	Missile 2 position	53254
HPOSM3	Missile 3 position	53255
SIZEP0	Player 0 size	53256
SIZEP1	Player 1 size	53257
SIZEP2	Player 2 size	53258
SIZEP3	Player 3 size	53259
SIZEM	Size of all missiles	53260

BASIC TUTORIAL - ANIMATION 4 continued

COLLISION REGISTERS (READ)
==========

REGISTER	MEANING	LOCATION
MODISP	Missile 0/Display	53248
M1DISP	Missile 1/Display	53249
M2DISP	Missile 2/Display	53250
M3DISP	Missile 3/Display	53251
PODISP	Player 0/Display	53252
P1DISP	Player 1/Display	53253
P2DISP	Player 2/Display	53254
P3DISP	Player 3/Display	53255
MOPL	Missile 0/Player	53526
M1PL	Missile 1/Player	53527
M2PL	Missile 2/Player	53528
M3PL	Missile 3/Player	53529
POPL	Player 0/Player	53260
P1PL	Player 1/Player	53261
P2PL	Player 2/Player	53262
P3PL	Player 3/Player	53263
HITCLR	Clear collisions	53278

PRIORITY TABLE (623)
==========

BIT 3 = 8	BIT 2 = 4	BIT 1 = 2	BIT 0 = 1
COLO	COLO	PCOLO	PCOLO
COL1	COL1	PCOL1	PCOL1
PCOLO	COL2	COLO	PCOL2
PCOL1	COL3	COL1	PCOL3
PCOL2	PCOLO	COL2	COLO
PCOL3	PCOL1	COL3	COL1
COL2	PCOL2	PCOL2	COL2
COL3	PCOL3	PCOL3	COL3
COL4 Background	COL4 Background	COL4 Background	COL4 Background

POWER USERS' CORNER

by John Kasupski,

Orignally of WNYAUG, and reprinted by OL' HACKERS AUG, NY
Published by TWAUG with permission by OL'HACKERS AUG. NY.

In this article, we will examine hardware and software that is popular among Atari 8-bit power users. A power user is anyone who has done one or more of the following to his or her Atari 8 BIT computer system:

Installed a RAM upgrade (256K or more) and USES the extra RAM (either as a RAMdisk, or for programming in BASIC XE, or for running utilities or applications that require the additional RAM)

Connected a hard drive.

Connected an MIO (or Black Box), one of which is basically necessary to add a hard drive anyway, but...

Anyone who has and uses one or more (or all!) of the above is a power user. Simple? Well, not really. You also need to have the proper software to let you take advantage of all this neat hardware. For most of us, this begins with the right DOS. By far the most popular DOS among power users is SpartaDOS, either the disk-based version (3.2) or the cartridge-based version (SpartaDOS X). It doesn't matter which version you use, as long as you've been weaned from the menu environment prevalent with most other DOS types such as DOS 2.0S/2.5, SmartDOS, MyDOS, etc.

There is other software available which takes advantage of power-user type hardware, and "Power Users' Corner" will examine the old and the new software, as well as the hardware itself. And, believe me, there is an unbelievable amount of software out there, if you know where to look.

But for this issue, we're going to look at a piece of hardware, because there might be people out there who are reading this and are feeling left out because they aren't power users, and we want them to be able to join the ranks. So, we're going to talk about the MIO.

HARDWARE REVIEW

MIO stands for Multi-I/O and was manufactured by ICD, Inc., a familiar name to power users since they also manufatured SpartaDOS, the U.S. Doubler upgrade for Atari 1050 drives, the R-Time 8 cart, and several other neat toys that frequently show up in a power user's computer room.

ICD unfortunately has discontinued its line of products for the 8 BIT, but the MIO can still be purchased as a used item (and now from FINE TOONED Engineering), and it is well worth the investment of from $125 to $200 you can expect to pay for one.

POWER USERS' CORNER continued

Why is a MIO worth that much money as a used item? The key is in what it adds to your system: self- refreshed RAM (either 256K or 1 megabyte, depending on the model you get), industry standard RS232 and parallel interfaces (read: printer port and modem port), the ability to reconfigure drives from an internal ROM-based menu, and perhaps best of all, a SASI/SCSI interface that allows you to connect a hard drive to your Atari.

The MIO has its own power supply and connects to the parallel bus interface on the 800XL/1200XL, or the cartridge ECI ports on the 130XE. For the 130XE you also need an adaptor board, which is about the size of a pack of cigarettes and has two cartridge ports on it. The MIO will connect directly to an XL without the adaptor. The 130XE adaptor for the MIO used to cost about $20.00 from ICD, I got mine along with my MIO so I can't tell you how much it costs used, however, if someone sells you a MIO and they have an adaptor, make sure you get it if you have a 130XE (or intend to get one!!)

Since the MIO has its own power supply, the contents of the MIO ram is not lost when you turn off your Atari. Only a total power failure (or turning the MIO itself off) will erase the MIO RAM's contents. You can leave the MIO on while the computer is off (the power consumption is negligible - only a watt or two in its active state), and the MIO's refresh oscillator will keep the contents of its RAM intact while the computer is off.

The MIO RAM can be used as a RAMdisk or as a printer spooler, or both. The 256K MIO can have either one 256K RAmdisk with no spooler, or use both the spooler and a RAMdisk, with the RAM divided between the two in 32K increments. The 1-Meg MIO allows partitioning the RAM into two or more RAMdisks, with or without a spooler.

The printer port allows connecting any Centronics parallel printer, which can be configured as either P1: or P2: as desired, and the RS-232 port allows connecting modems that use in RS-232 port, allowing the MIO to also take the place of an 850 Interface, P: R: connection, etc. The modem port can be configured as R1:, R2:, or R3: as desired.

The MIO's internal ROM includes a configuration menu that lets you set the parameters for the printer and modem ports and RAMdisk/spooler functions. It also lets you configure the system for up to eight disk drives, which can be floppy drives, hard drive partitions, or MIO RAmdisks. Any physical drive can be configured to respond as any logical drive number, and you can swap them around at will. This lets you boot from ANY drive in your system, including the MIO RAMdisk. You can easily swap drives around and boot from a floppy drive one time, a hard drive partition the next, the MIO RAMdisk another

POWER USERS' CORNER continued

time, and so on as needed by your system requirements.

As far as hard drives are concerned, the MIO menu lets you save the configuration that you've set up in the MIO Configuration menu to the physical hard drive on LUN 0,0 (if you don't know what this means, you needn't worry about it...though you can find out from your local users' group!), so that you can safely power down the MIO secure in the knowledge that the next time you boot your system, the MIO will load its own previously saved configuration right off of the hard drive during the boot process...saving you the trouble of having to re-enter all that information in the configuration menu.

The MIO can be used with a variety of disk operating systems, but SpartaDOS is the most highly recommended (either 3.2 or the X cart). The MIO will also work with the better alternatives such as MyDOS 4.5, BUT there is a reason why most power users prefer SpartaDOS...

This brings us to the SCSI/SASI port. In addition to all the above features, the MIO also serves as a host adaptor and interface that allows you to connect SCSI or SASI type hard disk drives to your Atari 8 BIT! There have been a few articles in club newsletters in the past extolling the virtues of having a hard drive connected to your Atari, and we will NOT repeat them all here. We will, however, point out that a hard drive represents perhaps the ULTIMATE addition to your system, providing unbeatable storage capacity and instant access to incredible amounts of software and data.

SOFTWARE REVIEW

There is also a ShareWare utility called MIO_INIT.COM written by LVAUG's Ed Bachman, that lets you save and load multiple MIO configurations, which you can then load from a SpartaDOS batch file. The latest version of MIO_INIT will load a new MIO configuration from a batch file WITHOUT terminating batch file execution, giving you the power to load from batch files, custom MIO configurations tailored to each application.

The value of this is not readily obvious to non-power users. MIO_INIT makes it possible to load one MIO configuration along with your word processor (setting up a print spooler, for example), but use an entirely different configuration with your term program (such as a large RAMdisk to save your capture buffer to). All this can be done from batch files, once you have saved the necessary configurations to your hard drive. You may never need to enter the MIO menu again!

CONCLUSION

There are countless MIO utilities available as P.D. or ShareWare. Ed Bachman has written several of these, and they're all useful items

POWER USERS' CORNER continued

to have around. If you use a MIO regularly then I suggest you get on the A.C.U.T.E. BBS and download Ed's utilities if you haven't obtained them already.

Ed's program range from the simple (a utility to call the MIO menu from the SpartaDOS Command line) to the complex (a HD directory formatter that automatically maps out any bad sectors on that partition by allocating them to a file called SECTORS.BAD), with a number of excellent utilities in between. I've found many of them useful.

Editor's Note: This POWER USER'S CORNER is NOT a proprietary column, and I do NOT expect to write every installment. In other words, others are welcome and encouraged to submit reviews, tips and tricks, and other articles pertaining to the subject of this feature. So, all you Power Users out there, fire up your word processors and WRITE!!!

WORD SEARCH

While playing round with the Word Puzzles disk from our PD library (TW418), I thought I'd put together a word search for this issue. We would be interested to hear what you think of it, and whether you would like more in future issues.

There are 15 game titles hidden in the word search, see how long it takes you to find them. The solution can be found elsewhere in the newsletter.

Y	X	N	A	M	P	M	U	J	J	M	H	H	S	A
Y	O	C	E	T	Z	A	B	Z	S	N	V	C	I	J
W	F	S	X	E	D	N	G	R	E	D	R	N	T	O
J	H	Q	R	E	E	A	B	Q	A	A	R	I	G	O
Z	O	O	B	V	A	Y	O	N	B	E	X	N	L	U
F	S	M	A	M	D	O	D	B	V	S	G	H	J	B
D	Z	A	L	I	L	V	L	A	G	E	Z	A	E	A
A	S	A	L	D	I	E	C	O	J	Z	I	R	G	R
T	E	F	B	N	N	S	I	R	T	E	T	D	A	N
X	I	O	L	I	E	T	C	M	V	J	K	B	P	A
A	N	Y	A	G	W	C	L	E	B	R	F	A	M	J
D	O	W	Z	H	Z	U	N	U	E	Q	O	L	A	N
J	O	M	E	T	G	E	O	H	K	L	B	L	R	I
Z	G	V	R	T	S	P	I	D	E	R	M	A	N	N
K	H	N	H	M	O	O	N	M	I	S	T	B	A	W

WORD SEARCH continued

Here are the 15 game titles to search for:

BALLBLAZER SPIDERMAN CAVERNIA DEADLINE HARDBALL
MIDNIGHT MOONMIST SCRABBLE JUMPMAN RAMPAGE
GOONIES TETRIS AZTEC NINJA KULT

1020 PLOTTER PEN-REFILL

The pens are cartridges (C) that have a metal reservoir (R) for the ink. This is closed by a color-coded disc (D) from the centre of the disc protrudes the pen (P), a thin metal cylinder. Taking this out and injecting ink in R is messy and as the total pen-length is critical, inadvisable.

As the ink is used, air replaces ink in R; and the ink especially in P dries. So soaking the pen in water, (destilled please), helps. Do this when following the next instructions.

Air expands on heating! Hold an old C in hot water; P-side up, air bubbles escape from the seam between R and D. Now grasp C with tweezers (its hot) and insert it P down in the refill ink.

N.B. The seam between R and D must be below the liquid level, it is through this seam that ink is sucked in as R cools and the air in it contracts.

Use the stopper of your ink bottle to hold the ink for recharging. Leave the C in position about 10 minutes to allow enough suction time. Pour what is left in the stopper back into the bottle.

INKS: experiment!But on principle reject all inks that show a precipitate ("sludge") on the bottom of the bottle. Excellent blue is HIGGINS DRAWING INK TG1 NR.450444 of FABER-CASTELL. Useable red is ROTRING VARIOGRAPH NR.590103.

Bill Walraven
P.Lammekensstraat 6
4827CM BREDA.THE NETHERLANDS.

TYNE&WEAR ATARI USER GROUP

T.W.A.U.G.
P.O.BOX No.8
WALLSEND
TYNE & WEAR
NE28 6DQ
Phone: 0191-2710086

CRACKING THE CODE

by Keith Mayhew

Re-printed by M. Gerum

This article first appeared in "The UK ATARI Computer Owners Club" later renamed "MONITOR"

Part 14

We start our tour of the Operating System (OS) with the Central Input Output (CIO) utility. CIO provides a consistent method of writing and reading data to and from devices such as the keyboard, disk drive, cassette or display.

OVERVIEW OF THE OPERATING SYSTEM

Before we delve into the mysteries of the CIO it is useful to gain an insight into the basic structure of the OS and the services it offers.

The CIO is at the top of a hierarchy of routines used to support input and output. There is one device driver routine for each physical type of device, these are either permanent or can be loaded at any time for additional devices, such as RS232 ports. To support devices which are attached via the serial port, such as a disk drive, the Serial Input Output (SIO) utility provides a high-level communications facility.

The parts of the OS which are not directly involved in input or output operations are actually rather minimal. They are the interrupt services and the so-called monitor which is executed at power-up and reset to initialise the system and pass control to another piece of software such as BASIC.

The floating point package (FPP), although part of the OS, is completely separate in its operation and is best thought of as a set of utility routines for your use in manipulating non-integer numbers.

The OS occupies the 8K bytes from E000 to FFFF in all systems, with the systems character set data taking the first 1K bytes from E000 to E3FF. The FPP resides adjacent to the OS in the 2K region from D800 to DFFF.

The XL & XE machines also have ROM in the 4K region from C000 to CFFF providing an additional character set from CC00 to CFFF. The extra ROM space has been used for the self-test procedure and other minor additions.

RAM USAGE

As you are no doubt aware, page zero of memory from 00 to FF is an important area of memory for any program using 6502: it gives greater speed, smaller code size and most significantly, the ability to perform indirection, i.e. a pair of locations can be used as a pointer.

The OS allocates the first 128 bytes of page zero for its own purposes leaving the other half from 80 to FF free for applications. However, if the floating point package is used , it makes active use of locations D4 to FF, leaving only 80 to D3 free. If you do not call any FPP routines then you are of course free to use them as you wish.

If a language such as BASIC is being run then it takes further locations. In the case of standard ATARI BASIC the following locations are used: 80 to CA and D2 to D3. This leaves a users program running under BASIC with access to locations CB to D1, as BASIC also uses the FPP. That gives just seven bytes which are safe to access!!!

A program, such as a game, which never uses the OS can use all the memory as it wishes. Other programs will have to be very careful otherwise they may cause unexpected results.!

Page one is always reserved for the stack, as dictated by the design of the 6502. If you plan to push a large number of values onto the stack then you should either test that the stack does not overflow (wrap around on itself) or set up your own private stack using a page zero pointer.

The memory from pages two to six inclusive are considered reserved and so programs should load from page seven onward.

Locations 200 to 47F are used by the OS leaving locations 480 to 6FF free for user's data. Well not quite! The FPP reserves the locations from 57E to 5FF. This leaves page 6 as the only completely unused page in lower

CRACKING THE CODE continued

memory.

OS VECTORS

The ATARI operating system makes extensive use of vectors for access to its resources. A vector being a pointer to a routine to be executed. By using this method, it means that only the vectors need be kept at fixed locations while the actual routines can reside anywhere. By adhering to the use of these vectors and other pointers means that programs will work under different revisions of the operating system.

The pointers in RAM are particularly useful as they can be "redirected" to your own routines, allowing you to extend or modify the function of the OS.

CENTRAL INPUT OUTPUT UTILITIES

The CIO is the most useful part of the IO system of the ATARI as it provides a consistent, mostly device independent, interface to any IO device. This means that to access different devices you do not need to know too much about the characteristics of any particular device.

In fact the CIO access method is essentially directly modelled in BASIC's file access commands such as OPEN, CLOSE, INPUT, PRINT, GET and PUT. All BASIC does is translate these commands into calls to the CIO.

CALLING THE CIO

All CIO calls are made via one vector at E456. The instruction at this address is a jump instruction to the actual CIO entry point. The data structures used to communicate between the CIO and the user's program are called IO Control Blocks. There are eight IOCBs of sixteen bytes each starting at 340 and extending to 3BF.

Each of these IOCBs hold information for communications via a "channel". Each channel is either open or closed and when open is performing either input, output or both to one and only one device. When one of the eight channels is closed any remaining data is transferred and the channel then becomes free for future use. Note that any channel can be assigned to any device: there is no difference between any of the eight channels.

Table 1 shows the allocation of the bytes within an IOCB and their descriptions. The most important of these are as follows: The function you wish the CIO to execute next for the appropriate channel is passed in ICCMD; Table 2 lists the available commands. ICSTA is the ststus of the last CIO command executed for the channel; Table 3 lists the status values. The user buffer which is used to hold the bytes to be transferred is pointed to by ICBAL & ICBAH, its length is held in ICBLL & ICBAH. ICAX1 through ICAX6 hold device dependent information.

The lists in Table 2 and Table 3 are the standard values which are implemented on all systems and include the basic disk operations which all versions of DOS support. New device drivers may add commands and new error codes to those given: refer to the appropriate manuals for further details.

To execute any CIO command the following needs to be done.

1. Fill in any necessary bytes in an IOCB.
2. Load the X register with the index of this IOCB from the location 340hex.
This is the equivalent to 16 times the channel number (0 to 7).
3. Jump to subroutine at E456hex.
4. If necessary, examine the Y register, which holds a copy of the status code stored in the IOCB, to determine if an error has occured. Note that all errors have bit 7 set, i.e. they are negative, so a BMI instruction will detect this upon return from CIO.

BASIC CIO COMMANDS

The first seven commands in Table 2 are the most common CIO functions which are supported by almost every device. A description of which follows:

OPEN

The OPEN ICOB command (03) has to be performed before most other CIO commands can take place

CRACKING THE CODE continued

on any particular IOCB.

The buffer address (ICBAL & ICBAH) must point at an ASCII string terminated with an end-of line (EOL) character of 155 decimal (EOL is also the character returned when pressing the RETURN key).

The ASCII string consists of an initial letter specifying the device, an optional number to specify one or several of those devices, e.g. two disk drives, a colon (:) and an optional filename for that device,e.g. a disk. This is in fact exactly the same as the names which are used in a BASIC filename. The following are the standard device names:

E: Editor.
S: Screen.
K: Keyboard.
C: Cassette.
P: Printer.
D: Disk drive, Not resident.
R: RS232. Not resident.

Note that the disk, i.e. DOS, and the RS232 devices are not resident in the computer and should be added to the system during power-up if they are required. It is also important to note that the editor is sometimes opened by the OS so that it can display messages or to prepare the screen ready for another application. IOCB number zero is always reserved by the OS for this purpose and it is best not to close and open it for devices other than "E:".

ICAX1 & ICAX2 are used to convey (device dependant) information on how to open the specified device. These two locations correspond to the numbers specified in a BASIC OPEN command. Usually zero will be placed in ICAX2 and ICAX1 will have 4 for read-only, 8 for write-only and 12 for read and write. ICAX3 through ICAX6 are used by very few devices and can be ignored in most cases.

CLOSE

The CLOSE IOCB command (0C) is used on an IOCB which has previously been opened and is no longer required. No extra information needs to be specified (not even a file name). A close will ensure that any remaining data is written to a device.

GET & PUT CHARACTERS

The GET characters command (07) reads the number of bytes specified by the buffer length (ICBLL & ICBLH) into the memory specified by the buffer address (ICBAL & ICBAH). The PUT characters command (0B) writes the specified number of bytes from the buffer to a device.

For either command a zero buffer length can be specified in which case the character is read into or written from the 6502's accumulator. All IOCB's values are left as they were before the call except that a GET characters command returns the number of bytes actually read in the buffer length locations. If fewer bytes have been read then it is because an end-of-file (EOF) has been encountered.

GET & PUT A RECORD

These two commands are similar to the previous two commands but these are useful when you do not know in advance exactly how many characters you wish to read or write. The GET record command (05) reads bytes into the specified buffer until an EOL character is encounted and written to the buffer. You still have to specify how long the buffer is and get the number of bytes actually read returned in ICBLL & ICBLH. If the buffer fills before an EOL is found then CIO reads the remaining characters from the device (ready for the next read) and returns an error code for a truncated record.

The PUT record command (09) writes all the characters in the specified buffer up to and including the first EOL character to the device. If the buffer does not contain an EOL then CIO will automatically write one to the device afterwards.

STATUS

All CIO calls return a status code in the Y register immediately after a call which can be tested to see if an error has occurred. In addition this value is saved in the status byte of the IOCB for later reference.

CRACKING THE CODE continued

The STATUS command (ØD) peforms more extensive status information which is device dependant. However all resident handlers do not support this function and simply return one in the Y register: the code for 'no error'.

AN EXAMPLE PROGRAM

Listing 1 is the assembly language code routine which prompts the user for an input and an output device/filename and copies from one to the other until end-of-file is reached. This performs exactly the same operation as the copy utility in DOS but is useful to illustrate how most of the CIO works and thus how to use it in your own programs. Listing 2 is the BASIC program to read the code into memory which is executed by typing:

X=USR(24576)

The program has several routines which provide the interface with the CIO routines. Each expects certain IOCB parameters to be set and the X register to contain the IOCB index, i.e. the IOCB channel numbers times sixteen. "OPEN" opens an IOCB but closes it first to ensure that it will be free. It assumes that the buffer address has been set to a filename string and that the auxiliary bytes have been set appropriately. "READLN" reads a record into the specified buffer. "WRITELN" writes a record from the specified buffer. The buffer length is always set to FFFF as the record should be followed by an EOL. "GETBYTE" reads a single character into the accumulator. No IOCB parameters need to be set. "PUTBYTE" writes a single character from the accumulator. No IOCB parameters need to be set. "GETCHRS" reads the specified number of characters into the given buffer. "PUTCHRS" writes the specified number of characters from the given buffer. "CLOSE" closes the IOCB. No IOCB parameters need to be set.

The program starts by re-opening the editor as the program cannot guarantee that it will always be open. It then prints a title message and prints a prompt for the input filename. A call is made to "INPUT" which reads a line of text typed at the keyboard via "READLN" which will be terminated with the EOL character, i.e. return.

A test is then made to see if "ERROR" needs to be called. This may be necessary in case the user pressed the break key which causes a CIO error. The specified filename is then opened for read-only and the process is repeated for the destination filename.

If the two files opened successfully then the file is copied by the routine "CPYFILE". This tries to read a buffer full of data from the first file. It then checks the status code. If it is not an error or it is end-of-file then it copies the buffer length, which indicates how many characters were actually read into the other IOCB and writes out the data.

After a write a further check is made to see if the previous read was at end-of-file. If it was not it jumps back to continue reading otherwise it returns to the user.

Any call to "ERROR" calls on "CIOERR" and then tries to close both the input and the output files. It then terminates the program by returning to the user. "CIOERR" is a useful piece of code which prints an error message and the error code in decimal. Printing the code in decimal is not as easy as it may at first seem: the obvious way is to divide by multiples of 10 and to take the remainders. This is however a rather messy solution and slow if you wanted to extend it to big numbers.

The solution adopted is quite elegant: it sets decimal mode with "SED" and then shifts the number to be converted one bit at a time to the left. On each shift the result bytes are doubled by adding each of them to itself, the first byte gets incremented if the carry was set from the shift operation. At the end of the loop, decimal mode is cancelled and the result contains the binary coded decimal (BCD) for the number. A simple printing routine then outputs each digit. A check is also made to suppress any leading zeros from the number to make the output neater.

It would be very easy to

CRACKING THE CODE continued

modify the decimal conversion routine for larger numbers making it of more general use for other programs, e.g. printing memory address.

The error handling is rather primitive in this example: it always terminates the program. It is usual for a program to effectively "trap" its errors causing some recovery action to be taken. For example, a writing error might ask the user to check the disk (if it was a disk drive) so that a retry can be made.

NEXT TIME

That completes the description of our example CIO program which should help you to write your own routines to perform general input or output. Next time we will look at how the CIO uses device drivers to implement the actual input or output to a device and the facilities each of the offers.

CIO Command bytes

Value	Operation
03	Open channel
05	Get record
07	Get characters
09	Put record
0B	Put characters
0C	Close channel
0D	Get status
11	Fill area
12	Draw line
20	Rename file
21	Delete file
22	Format disk
23	Lock file
24	Unlock file
25	Point in file
26	Note from file

Note: values 11 and 12 are for display only, and values 20 to 26 are for disk only.

Table 2. CIO Command Bytes

IOCB Locations

Location	Name	Description
IOCB+0	ICHID*	Handler ID
IOCB+1	ICDNO*	Device Number
IOCB+2	ICCMD	Command
IOCB+3	ICSTA	Status. Same as Y register after call
IOCB+4	ICBAL	Buffer address low & high
IOCB+5	ICBAH	
IOCB+6	ICPTL*	'Put' address
IOCB+7	ICPTH*	
IOCB+8	ICBLL	Buffer length low & high
IOCB+9	ICBLH	
IOCB+A	ICAX1	Auxiliary bytes
IOCB+B	ICAX2	
IOCB+C	ICAX3	
IOCB+D	ICAX4	
IOCB+E	ICAX5	
IOCB+F	ICAX6	

* Intended for internal use only.

Table 1. IOCB Locations

CIO Error Codes

Hex	Decimal	Description
1	1	Operation successful (no errors)
80	128	Break key abort
81	129	IOCB already open
82	130	Non-existent device name
83	131	Opened for writing only
84	132	Invalid command byte
85	133	Device/file not open
86	134	Invalid IOCB channel specified in Y register
87	135	Opened for reading only
88	136	End of file reached on read
89	137	Truncated record
8A	138	Device timeout
8B	139	Device NAK (negative acknowledge)
8C	140	Serial bus input framing error
8D	141	Cursor out of range
8E	142	Serial bus data frame overrun error
8F	143	Serial bus data frame checksum error
90	144	Device 'done' error
91	145	Invalid screen mode
92	146	Function not supported
93	147	Insufficient memory for screen mode
A0	160	Disk drive number invalid
A1	161	Too many open disk files
A2	162	Disk full
A3	163	Fatal disk IO error
A4	164	Internal file numbers inconsistent
A5	165	File name invalid
A6	166	'Point' to non-existent sector
A7	167	File 'locked'
A8	168	Command invalid for DOS
A9	169	Directory full
AA	170	File not found
AB	171	'Point' to position out of file

Table 3. CIO Error Codes

CRACKING THE CODE continued

```
0100 ;Example program using CIO to copy files.
0110 ;
0120 ;O.S. vectors...
0130 CIOV    =       $E456
0140 ;IOCB locations...
0150 ICCOM   =       $0342   ;Command code.
0160 ICBAL   =       $0344   ;Buffer address low.
0170 ICBAH   =       $0345   ;Buffer address high.
0180 ICBLL   =       $0348   ;Buffer length low.
0190 ICBLH   =       $0349   ;Buffer length high.
0200 ICAX1   =       $034A   ;Auxiliary 1.
0210 ICAX2   =       $034B   ;Auxiliary 2.
0220 ;IOCB commands...
0230 COPEN   =       $03     ;Open device.
0240 CGETREC =       $05     ;Get record.
0250 CGETCHR =       $07     ;Get characters.
0260 CPUTREC =       $09     ;Put record.
0270 CPUTCHR =       $0B     ;Put characters.
0280 CCLOSE  =       $0C     ;Close device.
0290 ;O.S. equates...
0300 RD      =       $04     ;Read from file.
0310 WR      =       $08     ;Write to file.
0320 EOL     =       $9B     ;End of line flag.
0330 EOF     =       $88     ;End of file error.
0340 ;Program equates.
0350 EDIOCB  =       $00     ;IOCB index for editor 'E: '.
0360 FMIOCB  =       $10     ;IOCB index for 'from' file.
0370 TOIOCB  =       $20     ;IOCB index for 'to' file.
0380 LNBUFL  =       $60     ;Length of line buffer.
0390 BUFLEN  =       $0400   ;Length of copy buffer.
0400 ;Page zero variables...
0410         *=      $CB
0420 STATUS  *=      *+1     ;Status of CIO command.
0430 ZERO    *=      *+1     ;Flag used when printing.
0440         *=      $6000
0450         PLA
0460         LDX     #EDIOCB ;Open editor in case closed.
0470         LDA     #EFILE&$FF ;File spec. pointer.
0480         STA     ICBAL,X
0490         LDA     #EFILE/256
0500         STA     ICBAH,X
0510         LDA     #RD+WR  ;Read and write.
0520         STA     ICAX1,X
0530         JSR     OPEN
0540         LDA     #MSTITLE&$FF ;Print title string.
0550         STA     ICBAL,X
0560         LDA     #MSTITLE/256
0570         STA     ICBAH,X
0580         JSR     WRITELN
0590         LDA     #EOL    ;Skip two lines.
0600         JSR     PUTBYTE
0610         JSR     PUTBYTE
0620         LDA     #MSFROM&$FF ;Message for 'from' file.
0630         STA     ICBAL,X
0640         LDA     #MSFROM/256
0650         STA     ICBAH,X
0660         JSR     WRITELN
0670         JSR     INPUT   ;Get input line.
0680         BMI     ERROR   ;Catch errors e.g. 'BREAK'.
0690         LDX     #FMIOCB ;Try to open given file name
0700         LDA     #RD     ;for reading.
0710         JSR     OPNFILE
0720         BMI     ERROR
0730         LDX     #EDIOCB ;Message for 'to' file.
0740         LDA     #MSTO&$FF
0750         STA     ICBAL,X
0760         LDA     #MSTO/256
0770         STA     ICBAH,X
0780         JSR     WRITELN
0790         JSR     INPUT
0800         BMI     ERROR
0810         LDX     #TOIOCB ;Try to open file.
0820         LDA     #WR
0830         JSR     OPNFILE
0840         BMI     ERROR
0850         JSR     CPYFILE ;Both files open: start copy.
0860         LDX     #FMIOCB ;Close both files.
0870         JSR     CLOSE
0880         LDX     #TOIOCB
0890         JSR     CLOSE
0900         RTS             ;All done...
0910 ;Handle a CIO error...
0920 ERROR   STY     STATUS  ;Save status code.
0930         JSR     CIOERR  ;Print error message.
0940         LDX     #FMIOCB ;Make sure both files
0950         JSR     CLOSE   ;are closed.
0960         LDX     #TOIOCB
0970         JSR     CLOSE
0980         RTS
0990 ;Copy 'from' file to 'to' file...
1000 CPYFILE LDX     #FMIOCB ;'from' file buffer address.
1010         LDA     #BUFFER&$FF
1020         STA     ICBAL,X
1030         LDA     #BUFFER/256
1040         STA     ICBAH,X
1050         LDA     #BUFLEN&$FF ;Set length.
1060         STA     ICBLL,X
1070         LDA     #BUFLEN/256
1080         STA     ICBLH,X
1090         LDX     #TOIOCB ;'to' file buffer address.
1100         LDA     #BUFFER&$FF
1110         STA     ICBAL,X
1120         LDA     #BUFFER/256
1130         STA     ICBAH,X
1140 READNXT LDX     #FMIOCB ;Read a block into buffer.
1150         JSR     GETCHRS
1160         STY     STATUS  ;Save status.
1170         BPL     WRTBUF  ;If OK then write buffer.
1180         CPY     #EOF    ;If not EOF then error.
1190         BNE     ERROR
1200 WRTBUF  LDY     #FMIOCB ;Copy length of block.
1210         LDX     #TOIOCB
1220         LDA     ICBLL,Y
1230         STA     ICBLL,X
1240         LDA     ICBLH,Y
1250         STA     ICBLH,X
```

CRACKING THE CODE continued

```
1260          JSR     PUTCHRS ;Write block.
1270          BMI     ERROR
1280          LDA     STATUS  ;Stop if EOF from last read.
1290          BPL     READNXT ;Else do next block.
1300          RTS             ;All copied.
1310 ;Open file specified by input buffer.
1320 OPNFILE STA      ICAX1,X ;Save read/write info.
1330          LDA     #$00    ;Zero auxiliary 2.
1340          STA     ICAX2,X
1350          LDA     #LINBUF&$FF
1360          STA     ICBAL,X
1370          LDA     #LINBUF/256
1380          STA     ICBAH,X
1390          JSR     OPEN
1400          RTS
1410 ;Get user input into line buffer.
1420 INPUT    LDA     #LINBUF&$FF
1430          STA     ICBAL,X
1440          LDA     #LINBUF/256
1450          STA     ICBAH,X
1460          LDA     #LNBUFL&$FF
1470          STA     ICBLL,X
1480          LDA     #LNBUFL/256
1490          STA     ICBLH,X
1500          JSR     READLN
1510          RTS
1520 ;Open a file, closing IOCB first.
1530 OPEN     JSR     CLOSE
1540          LDA     #COPEN
1550          STA     ICCOM,X
1560          JSR     CIOV
1570          RTS
1580 ;Read up to end of line.
1590 READLN   LDA     #CGETREC
1600          STA     ICCOM,X
1610          JSR     CIOV
1620          RTS
1630 ;Write up to end of line.
1640 WRITELN LDA      #CPUTREC
1650          STA     ICCOM,X
1660          LDA     #$FF    ;Maximum buffer length
1670          STA     ICBLL,X ;as string is terminated
1680          STA     ICBLH,X ;by EOL character.
1690          JSR     CIOV
1700          RTS
1710 ;Get a single byte in accumulator.
1720 GETBYTE LDA      #CGETCHR
1730          STA     ICCOM,X
1740          LDA     #$00    ;Zero buffer length.
1750          STA     ICBLL,X
1760          STA     ICBLH,X
1770          JSR     CIOV
1780          RTS
1790 ;Put a single byte from accumulator.
1800 PUTBYTE PHA
1810          LDA     #CPUTCHR
1820          STA     ICCOM,X
1830          LDA     #$00    ;Zero buffer length.
1840          STA     ICBLL,X
1850          STA     ICBLH,X
1860          PLA
1870          JSR     CIOV
1880          RTS
1890 ;Get specified number of characters.
1900 GETCHRS LDA      #CGETCHR
1910          STA     ICCOM,X
1920          JSR     CIOV
1930          RTS
1940 ;Put specified number of characters.
1950 PUTCHRS LDA      #CPUTCHR
1960          STA     ICCOM,X
1970          JSR     CIOV
1980          RTS
1990 ;Close file.
2000 CLOSE    LDA     #CCLOSE
2010          STA     ICCOM,X
2020          JSR     CIOV
2030          RTS
2040 ;Print CIO error status.
2050 CIOERR   SED             ;Decimal mode.
2060          LDA     #$00    ;Zero result.
2070          STA     NUMBER
2080          STA     NUMBER+1
2090          LDX     #8      ;Eight bits.
2100 NEXTBIT ASL      STATUS  ;Get next bit in carry.
2110          LDY     #1      ;Double number adding carry.
2120 ADDLOOP LDA      NUMBER,Y
2130          ADC     NUMBER,Y
2140          STA     NUMBER,Y
2150          DEY
2160          BPL     ADDLOOP ;Next byte.
2170          DEX
2180          BNE     NEXTBIT ;Next bit.
2190          CLD             ;Back to binary mode.
2200          JSR     PRNTERR ;Print error message.
2210          RTS
2220 ;Print CIO error message and number.
2230 PRNTERR LDX      #EDIOCB ;Print message.
2240          LDA     #MSERROR&$FF
2250          STA     ICBAL,X
2260          LDA     #MSERROR/256
2270          STA     ICBAH,X
2280          LDA     #MSERRL&$FF
2290          STA     ICBLL,X
2300          LDA     #MSERRL/256
2310          STA     ICBLH,X
2320          JSR     PUTCHRS
2330          LDA     #$00    ;Clear 'zero' flag.
2340          STA     ZERO
2350          LDA     NUMBER  ;Start with third digit.
2360          AND     #$0F
2370          STA     ZERO    ;Save.
2380          BEQ     MDDIGIT ;Don't print if zero.
2390          JSR     PUTNUM
2400 MDDIGIT LDA      NUMBER+1 ;Second digit.
2410          JSR     GETHIGH
```

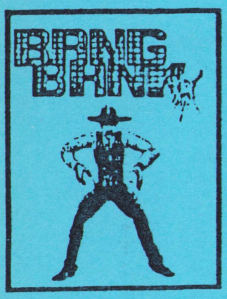
CRACKING THE CODE continued

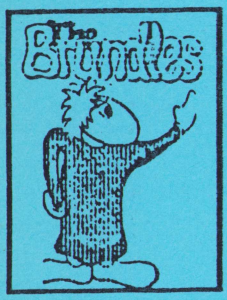
```
2420          TAY              ;Save.
2430          ORA      ZERO    ;OR to previous digit.
2440          STA      ZERO
2450          BEQ      LODIGIT ;Skip if a leading zero.
2460          TYA              ;Print digit.
2470          JSR      PUTNUM
2480 LODIGIT LDA       NUMBER+1 ;Print first digit.
2490          AND      #$0F
2500          JSR      PUTNUM
2510          LDA      #EOL    ;Print end of line.
2520          JSR      PUTBYTE
2530          RTS
2540 ;Return high nibble of 'A' in low nibble.
2550 GETHIGH LSR       A
2560          LSR      A
2570          LSR      A
2580          LSR      A
2590          RTS
2600 ;Print a single digit.
2610 PUTNUM  CLC
2620          ADC      #'0'
2630          JSR      PUTBYTE
2640          RTS
2650 ;Messages...
2660 MSTITLE .BYTE     "COPY FILE",EOL
2670 MSFROM  .BYTE     "From file?",EOL
2680 MSTO    .BYTE     "To file?",EOL
2690 MSERROR .BYTE     "Error number "
2700 MSERRL  =         *-MSERROR ;Length of message.
2710 EFILE   .BYTE     "E:",EOL  ;Editor file spec.
2720 NUMBER  *=        *+2       ;Error number in decimal.
2730 LINBUF  *=        *+LNBUFL  ;Line buffer.
2740 BUFFER  *=        *+BUFLEN  ;Copy buffer.
```

Listing 1.

```
10 DIM HEX$(16)
20 LINE=10000:TRAP 100:J=0:START=24576
30 READ HEX$,CHKSUM:SUM=0
40 FOR I=1 TO 15 STEP 2
50 D1=ASC(HEX$(I,I))-48:D2=ASC(HEX$(I+1,I+1))-48
60 NUM=((D1-7*(D1>16))*16+(D2-7*(D2>16)))
70 SUM=SUM+NUM:POKE START+J,NUM:J=J+1:NEXT I
80 IF SUM=CHKSUM THEN LINE=LINE-10:GOTO 30
90 ? "Checksum error on this line!"
95 LIST LINE:END
100 PRINT "Data in memory."
10000 DATA 68A200A9FE9D4403,817
10010 DATA A9619D4503A90C9D,333
10020 DATA 4A0320F760A9D39D,989
10030 DATA 4403A9619D450320,598
10040 DATA 0C61A99B202E6120,640
10050 DATA 2E61A9DD9D4403A9,930
10060 DATA 619D450320C6120,499
10070 DATA DF60303 4A210A904,770
10080 DATA 20C960302BA200A9,751
10090 DATA E89D4403A9619D45,952
10100 DATA 8320C6120DF6203,543
10110 DATA 17A220A9B820C960,723
10120 DATA 300E20B060A21020,528
10130 DATA 5361A2202536160,682
10140 DATA 84CB205C61A21020,766
10150 DATA 5361A2202536160,682
10160 DATA A210A9B39D4403A9,875
10170 DATA 629D4503A9009D48,725
10180 DATA 03A9049D4903A220,603
10190 DATA A9839D4403A9629D,952
10200 DATA 4503A21020416184,576
10210 DATA CB1004C068D0C1A0,1112
10220 DATA 10A220B94B039D48,699
10230 DATA 03B949039D490320,529
10240 DATA 4A6130ACA5CB10DA,993
10250 DATA 609D4A03A9009D4B,731
10260 DATA 03A9039D4403A962,670
10270 DATA 9D450320F76060A9,869
10280 DATA 039D4403A9629D45,724
10290 DATA 03A9B09D4B03A900,701
10300 DATA 9D49032003616020,493
10310 DATA 5361A9039D420320,610
10320 DATA 56E460A9059D4203,810
10330 DATA 2056E460A9099D42,843
10340 DATA 03A9FF9D4B039D49,889
10350 DATA 032056E460A9079D,778
10360 DATA 4203A9009D4B039D,527
10370 DATA 49032056E46048A9,739
10380 DATA 8B9D4203A9009D48,635
10390 DATA 039D49036820 56E4,686
10400 DATA 60A9079D42032056,516
10410 DATA E460A90B9D420320,762
10420 DATA 56E460A90C9D4203,317
10430 DATA 2056E460FBA9008D,1000
10440 DATA 0162BD0262A20806,516
10450 DATA CBA001B901627901,770
10460 DATA 62990162BB10F4CA,948
10470 DATA D0EDDB207F6160A2,1175
10480 DATA 00A9F19D4403A961,904
10490 DATA 9D4503A90D9D4803,643
10500 DATA A9009D490320 4A61,605
10510 DATA A90085CCAD016229,819
10520 DATA 0FB5CCF00320CC61,928
10530 DATA AD026220C761A805,774
10540 DATA CCB5CCF004982 0CC,1173
10550 DATA 61AD0262290F20CC,662
10560 DATA 61A99B202E61604A,766
10570 DATA 4A4A4A6018693020,527
10580 DATA 2E6160434F505920,586
10590 DATA 46494C459B46726F,738
10600 DATA 6D2066696C653F9B,775
10610 DATA 546F2066696C653F,706
10620 DATA 9B4572726F72206E,819
10630 DATA 756D6265722045 3A,698
10640 DATA 9B,155
```

Listing 2.

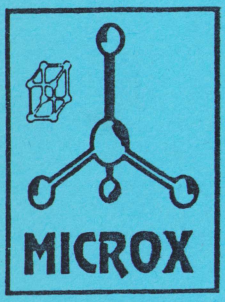




Are You Still Complaining about lack of New software for your Atari 8 bit ?
If you are you don't know about Mail Order From

MICRO-DISCOUNT

265, CHESTER ROAD.
STREETLY.
WEST MIDLANDS.
B74 3EA.
ENGLAND.



TEL 021 353 5730
FAX 021 352 1669

ADAX *HANS KLOSS*

DARKNESS HOUR

The Atari Classic Programmer's Club

This is to inform you membership is now only available on yearly subscriptions. The life membership option is no longer available. A member has access to various programmer services, a helpline, regular printed newsletter and discounts off our software?

Current UK membership cost is:

12 months: £6.00

Overseas members most welcome. If you require more details then please send an SAE to the address below. Overseas Atarians, please send two International Reply Coupons (available from your post office) for more information.

Still Available:

Swift Spreadsheet (Standard) - £9.95

Swift Spreadsheet (New) - £12.95

(NOTE: New version includes a revised 40 page A4 manual. For details of the standard version and of the program itself, please refer to the review in issue 67 of New Atari User).

Available Soon:

Menu Print: Still under development and currently undergoing its THIRD rewrite! We apologise for the delay but we expect the programming to be definitely completed by the end of September. More details will be published when ready.

THE ATARI CLASSIC PROGRAMMER'S CLUB
Pen-Tyddyn
Capel Coch
LLangefni
Anglesey, Gwynedd LL77 7UR
Wales

CURRENT NOTES

Helping Atari Owners Through the World of Computing.

Current Notes is published bi-monthly in the U.S.

This magazine has 80 pages full of computer news and very good articles that covers the Atari 8-bit and ST. It comes in full size of 11 inches by 8 1/2 inches.

SUBSCRIPTIONS:

Europe subscriptions is $69 per year.
Bankers drafts made payable to Current Notes

to: CN Subscriptions
122 N. Johnson RD.
Sterling, VA 20164

NOTE: VISA and MasterCard accepted.
Call (703) 450-4761

Editor's Note:
TWAUG is receiving the CN magazine and are always looking forward to it.

PHOENIX.

The new disk based news letter from Ireland, produced by Robert Paden.

PHOENIX a double sided disk, side 'A' will be packed full of text files containing Articles, reviews and much much more. Side 'B' will contain a good selection of PD software.

PHOENIX is only available from Robert Paden himself.